

시간 비밀 채널을 제거하는 실시간 트랜잭션 스케줄링 기법 (Real-Time Transaction Scheduling Method without Timing Covert Channel)

조숙경(Cho Sook Kyoung)¹⁾, 김종훈(Kim Jong Hoon)²⁾, 정현민(Jung Hyun Min)³⁾
정미영(Jung Mi Young)⁴⁾, 배영호(Bae Young Ho)⁵⁾, 배해영(Bae Hae Young)⁶⁾

요약

본 논문에서는 강제적 접근 제어를 구현한 보안 데이터베이스 시스템에서 실시간 트랜잭션을 효율적으로 처리하기 위한 스케줄링 기법을 제안한다. 제안된 기법에서는 시간 비밀 채널의 발생을 방지하고자 보안 등급에 기초한 대기 큐와 시간 제약 조건을 만족시키는 트랜잭션의 수를 최대로 하기 위해 트랜잭션의 타입, 종료시한, 중요도를 고려한 우선순위 큐를 이용한다.

제안된 기법은 보안등급이 다른 트랜잭션들 간의 불간섭 성질을 유지시켜 시간 비밀 채널을 제거하였으며, 종료시한을 만족시키는 실시간 트랜잭션의 중요도의 합을 최대로 하는 장점을 가진다. 또한, 제안된 기법은 성능평가를 통해서 기존의 기법에 비해 트랜잭션의 종료시한 만족 비율이 30% 정도 향상됨을 보였다.

Abstract

In this paper, we propose a **new scheduling method without timing covert channel of real-time transaction** for secure database systems that implement mandatory access control. Our scheduling method use the wait queue based on security level to remove timing covert channel. And it use priority queue that consider transaction type, deadline, and weight.

Therefore, the proposed scheduling method prevents timing covert channel because it is kept noninterference between transactions with different security level, and maximizes the sum of the weight of transactions that satisfy its deadline. The simulation results, is a comparison of traditional methods, show that our scheduling method is improved to 30%.

1) 정회원 : 인천대학교 강의전담교수

2) 정회원 : (주)KGI 대표이사

3) 정회원 : (주)케이티 서비스개발연구소 무선망설계 연구실장

4) 정회원 : (주)케이티 서비스개발연구소 선임보안연구원

5) 정회원 : (주)케이티 서비스개발연구소 선임보안연구원

6) 정회원 : 인하대학교 교수

논문접수 : 2003. 12. 15.

심사완료 : 2003. 12. 23.

1. 서론

트랜잭션이 종료시한(deadline)과 같은 시간 제약 조건(timing constraints)을 가지는 실시간 데이터베이스 시스템(real-time database systems)[1, 2]은 공장 자동화 시스템, 원자력 발전소 제어 시스템, 기상 인공위성 시스템, 미사일 제어 시스템과 같은 특정 응용 분야에서 주로 사용되었으나, 최근에 와서는 컴퓨터의 급속한 발전과 통신 기술의 눈부신 발전에 힘입어 ITS(Intelligence Transport Systems), 응급 의료 정보 서비스(Emergency Medicine Information Systems), 국방 통제 시스템(Military Control Systems), 전자상거래(Electronic Commerce), 유무선 통신, 멀티미디어 등과 같이 폭넓고 다양한 분야로 급격히 확대되고 있다[3]. 이러한 실시간 데이터베이스 시스템에서 발생하는 트랜잭션은 논리적으로 정확하게 처리되어야 할 뿐만 아니라 부여된 종료시한 내에 처리가 완료되어야만 한다. 즉, 국방 통제 시스템에서 항적에 대한 인지와 추적 그리고 제어를 포함하는 데이터베이스 연산은 어떤 것이든 실시간 요구 사항인 시간 제약 조건을 만족해야 한다[4, 5].

한편, 최근 인터넷과 컴퓨터의 보급이 증가하면서 통신, 운송, 국방, 금융 등의 국가 주요 기반 시설들이 정보 시스템에 의존하고 있는 이 때 정보 시스템에 대한 불법 침입, 데이터 변조, 데이터 삭제, 서비스 거부 등의 보안을 위협하는 행위가 빈번하게 일어나고 있으며, 이것은 국가 안보와 경제에 엄청난 피해를 초래할 수 있다. 더욱이 최근 인터넷 기업의 폭발적인 증가와 LAN 만으로 구성된 폐쇄망들이 정보 교환, 전자상거래, 은행 업무 등 각종 서비스 제공을 위해 데이터베이스 시스템을 인터넷에 연결하려는 시도들이 가속화되고 있는 지금 보안에 대한 중요성은 점점 더 강조되고 있는 실정이다[6]. 데이터베이스 보안의 목적은 권한이 없는 사용자를 제어하여 불법적인 접근, 고의적인 파괴 및 변경을 방지하고 우발적인 사고로부터 데이터를 보호하는데 있다. 즉, 데이터베이스 관리 시스템은 데이터베이스에 저장되어 있는 데이터에 대한 불법적인 접근, 고의적인 파괴, 변경 그리고 비밀관성을 발

생시키는 접근으로부터 데이터를 보호하기 위한 보안 정책을 수행하여야 한다[7, 8].

보안 데이터베이스 시스템들에서 실시간 기능을 지원해야 한다는 것은 기존의 데이터베이스 시스템에서와 마찬가지로 매우 중요하고 기본적인 문제로 인식되고 있다. 예를 들어, 비행기의 이착륙과 비행기에 대한 정보를 가지고 있는 관제탑에서 여객기의 비상 사태 메시지를 받으면 관제탑은 제한 시간 내에 여객기가 착륙할 수 있는 비행장을 찾기 위해 전국의 모든 비행장 정보를 가지고 있는 국방부에 도움을 청한다. 비행장 정보 중 일반 공항의 위치는 모두가 알아도 되는 데이터이지만 군용 비행장의 위치는 기밀 데이터라고 할 때 국방부는 여객기에게 시간 제약 조건 내에 착륙할 수 있는 일반 공항의 위치를 알려준다. 그러나, 군용기가 비상 사태에 직면해 5분내 착륙 가능한 비행장을 찾는다면 일반 공항은 물론 군용 비행장의 위치도 알려주게 된다. 이와 같이 질의를 요청한 사람의 보안등급과 접근되는 데이터의 보안등급에 따라 결과 값을 다르게 주어 기밀 데이터의 보안을 유지하며, 제한 시간이라는 시간 제약 조건 내에 결과 값을 알려주어 그 값이 가치를 유지할 수 있게 만들어 주는 시스템을 보안 실시간 데이터베이스 시스템(Secure Real-Time Database Systems)이라고 한다. 즉, 보안 데이터베이스 시스템에 실시간 특성을 지원하도록 하여 시간 제약 조건을 만족시키는 시스템을 보안 실시간 데이터베이스 시스템(Secure Real-Time Database Systems)이라고 한다.

지금까지 실시간 데이터베이스 시스템과 보안 데이터베이스 시스템에 대한 연구는 각각 별도로 이루어져 왔다. 즉, 실시간 데이터베이스 시스템에서는 종료시한과 관계된 시간적 제약 조건을 고려한 연구가 주로 이루어졌으며, 보안 데이터베이스 시스템에서는 비밀성과 무결성에 관계된 보안 특성을 고려한 연구로 독립적으로 진행되어 왔다.

보안 데이터베이스 시스템에 관한 연구는 대부분이 다단계 보안 데이터 모델에 관한 연구로서, 다양한 권한을 갖는 사용자들과 다양한 접근 등급을 갖는 데이터를 제어하는 강제

적 접근 제어(Mandatory Access Control : MAC)를 지원하는 다단계 보안 데이터베이스 시스템에 연구가 집중되고 있다. 강제적 접근 제어는 객체(object)에 포함된 정보의 비밀 등급(classification)과 주체에 부여된 비밀 취급 인가(clearance)를 기반으로 객체에 대한 접근을 제어하는 방식이다. 이는 다단계 보안 시스템의 주된 제어 방식으로 각 시스템의 주체와 객체에 보안 등급을 부여하고, 등급별로 분리된 정보가 하위 등급으로 흘러 내려가는 것을 방지하는 보안 시스템이다[7, 8]. 예를 들면 미국방부의 강제적 보안 정책은 인가된 개인에 따라 비밀 정보의 접근을 제한한다. 기밀 데이터는 비권한 사용자로부터 직접 접근을 보호해야 할 뿐만 아니라 추론과 같은 간접 수단을 통한 유출을 보호하는 것이 강제적 접근 제어에서는 요구된다. 강제적 보안은 다단계 보안(multi-level security)을 구현하기 위한 방법론의 핵심이 된다. 일반적으로 다단계 보안 데이터베이스 시스템에서는 Bell-LaPadula 모델[7, 8]에 따른 접근 제어 매커니즘을 사용한다.

이러한 다단계 보안을 위한 데이터베이스 시스템에서 실시간 특성을 지원할 때 발생하는 문제점은 다음과 같다. 발생하는 트랜잭션은 모두 특정 보안등급을 가지며, 그 중 실시간 트랜잭션은 시간 제약 조건인 종료시한을 가진다. 이것은 트랜잭션 스케줄링시 보안등급과 종료시한을 모두 고려해야 한다는 것을 의미한다. 보안등급을 기준으로 스케줄링을 하면 긴급한 트랜잭션이 종료시한을 지키지 못할 수 있고, 종료시한을 기준으로 스케줄링을 하면 시간 비밀 채널[9, 10]이 생성되어 기밀 데이터가 유출될 수 있다. 이것은 트랜잭션이 데이터에 접근할 때에는 Bell-LaPadula 모델의 제약 조건인 단순 속성과 제한된 *-속성을 따르기 때문이다. 단순 속성은 트랜잭션의 보안등급이 데이터의 보안등급보다 동일하거나 높은 경우에만 판독 연산을 허용하는 것이며, 제한된 *-속성은 트랜잭션의 보안등급이 데이터의 보안등급과 동일한 경우에만 갱신 연산을 허용하는 것이다. Bell-LaPadula 모델의 두 가지 속성은 허가되지 않은 사용자에게 직접적으로 기밀 데이터를 유출하는 것은 막을 수 있으나

간접적인 비밀 누출을 통한 불법적인 정보의 흐름인 비밀 채널(covert channel)을 방지할 수 없다[11]. 그래서, 비밀 채널을 방지하기 위해 낮은 보안등급 트랜잭션을 먼저 실행하는 정책을 사용한다. 그러나, 이것은 실시간 트랜잭션의 종료시한을 만족시키지 못하는 결과를 가져올 수 있으며 우선순위 역전 현상을 발생시킬 수 있다. 우선순위 역전은 높은 우선순위 트랜잭션이 낮은 우선순위 트랜잭션 때문에 실행이 지연되는 것으로서 실시간 데이터베이스 시스템에서는 방지하여야 하는 현상이다.

위와 같은 문제점들을 해결하기 위해서는 기존의 데이터베이스 시스템에서 개발된 트랜잭션 스케줄링 기법을 보안 정책과 실시간 특성에 따라 변경 적용할 수 있는 트랜잭션 관리 기법에 대한 연구가 필요하다. 이를 위한 본 논문의 구성은 다음과 같다. 2장에서 보안 데이터베이스 시스템의 특성과 기존의 실시간 트랜잭션 스케줄링 기법에 대하여 살펴보고, 3장에서는 트랜잭션의 보안등급에 기초한 대기 큐와 실시간 트랜잭션의 타입, 종료시한, 중요도를 고려한 우선순위 큐를 이용한 스케줄링 기법을 설명하며, 4장에서는 제안된 기법에 대한 성능평가를 수행하고, 5장에서는 마지막으로 본 논문에 대한 결론과 향후 연구 방향에 대하여 논한다.

2. 관련연구

2.1 보안 데이터베이스 시스템

대부분의 상용 데이터베이스 시스템들이 채택하고 있는 보안 유지 방법은 데이터에 대한 사용자들의 사용 권한을 제어하는 임의적 접근 제어(DAC : Discretionary Access Control) 방식들이다[7]. 임의적 접근 제어 방식은 보안 유지에 취약점을 드러내고 있는데 이는 데이터에 대한 사용 권한을 사용자 임의대로 다른 사용자들에게 양도할 수 있는 접근 제어 방식이기 때문이다. 이러한 임의적 접근 제어 방식은 대부분의 정직한 내부 사용자들에 대한 정보의 누출을 방지하는 경우에는 적합할 수 있으나, 악의적인 침입자들이 트로이 목마[7, 8]를 이용한 데이터의 접근 또는 컴퓨터 바이러스에 의한 데이터의 접근은 반드시 제한되

고 방지되어야 함에도 불구하고 원천적으로 방지할 수 없는 결함을 가지고 있다. 트로이 목마는 프로그램 내에 인가되지 않은 사용자에게 정보를 누출시키는 악의적인 코드를 수록한 것이다. 예를 들어, 유틸리티 프로그램 중에서 정렬 프로그램에 트로이 목마가 감추어져 있다면 사용자가 정렬 프로그램을 불러 자신의 파일을 정렬시키고자 할 때마다 인가받지 못한 사용자에게 자신의 파일들을 복사시킬 수 있는 것이다.

따라서, 임의적 접근 제어 방식의 결점을 극복하기 위한 강제적인 접근 제어(MAC : Mandatory Access Control) 방식이 개발되었다. 강제적 접근 제어 방식은 주체와 객체라는 용어에 의해 기술된 Bell-LaPadula 모델에 기초한다. 객체는 데이터 파일, 레코드 또는 레코드 내의 필드로 이해될 수 있으며, 주체는 객체들에 대한 접근을 요청할 수 있는 활성화된 프로세스이다. 모든 객체는 비밀 등급이 할당되며, 각각의 주체도 등급별 비밀 취급 인가가 결정되어야 한다.

2.1.1 Bell-LaPadula 모델

다단계 보안 모델에서는 모든 데이터와 사용자에게 보안등급(security level)을 부여한다. 데이터베이스 시스템이 다양한 보안등급을 가진 정보를 포함하고 가장 높은 보안등급의 데이터를 접근이 불가능한 사용자가 있을 때 다단계의 보안 필요성이 있다. 보안 등급은 $Unclassified(U) < Confidential(C) < Secret(S) < Top Secret(TS)$ 로 나누어진다. 사용자에게 할당된 보안등급은 기밀 정보를 유출하지 않을 사용자의 신뢰도를 반영한다. 다단계 데이터베이스 시스템은 다양한 보안등급을 가지는 데이터와 인가등급을 가지는 사용자를 유지하여야 한다. 가장 일반적인 경우라면 데이터베이스의 원자적 사실(fact)에 각각 보안등급을 부여하는 것이다. 이는 데이터의 접근등급과 접근을 요청하는 주체의 접근권한에 따라 데이터의 직접 혹은 간접적인 접근을 통제하는 능력을 의미한다. 이 요구 사항은 다음과 같이 형식화될 수 있다. 우선, 모든 가능한 접근 등급의 집합은 지배(dominant)라고 불리는

부분 순서 관계(partial ordering relation) \geq 를 갖는 래티스(lattice)로써 구조화된다. 그리고 각 주체 s 는 읽기를 위한 판독 권한 $read-class(s)$ 와 쓰기를 위한 기록 권한 $write-class(s)$ 의 두 가지 접근등급을 갖는다. 이 때 $read-class(s) \geq write-class(s)$ 의 지배 관계가 성립된다. 마지막으로, 인가되지 않는 노출과 파괴로부터 데이터를 보호하기 위하여 강제적 접근 제어의 요구 조건은 두 가지 규칙으로 형식화된다. 이는 Bell-LaPadula의 강제적 보안 정책에 기반을 두고 있다[7].

- 1) 단순 보안 속성 : 주체 s 는 접근등급 x 를 갖는 데이터에 대하여 $read-class(s) \geq x$ 의 관계가 성립되지 않는 한 데이터를 읽을 수 없다.
- 2) 제한 *-속성 : 주체 s 는 접근등급 x 를 갖는 데이터에 대하여 $write-class(s) = x$ 의 관계가 성립되지 않는 한 데이터를 기록할 수 없다.

첫 번째 성질은 사용자보다 높은 보안등급을 가진 데이터에 대한 접근을 방지하기 위한 것이다. 두 번째 성질은 높은 보안등급을 가진 객체의 정보를 낮은 보안등급을 가진 객체에 옮김으로써 일어나는 잘못된 정보 흐름(information flow)을 방지하기 위한 것이다. 즉, 두 가지 규칙에서 전자는 “상향 판독 금지(no read up)”, 후자는 “하향 기록 금지(no write down)” 규칙을 의미한다. 본 규칙을 데이터베이스 환경에서 준수하기 위해서는 SeaView 모델[7, 8]에서 소개한 다중인스턴스화(polyinstantiation)기법을 사용한다.

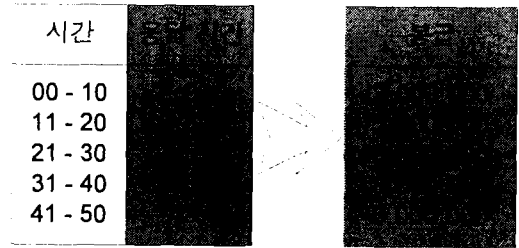
2.1.2 비밀 채널

Bell-LaPadula 모델의 단순 속성과 제한된 *-속성은 상위 보안등급 주체로 부터 하위 보안등급 주체에게로 정보 누출이 발생하지 않도록 고안한 것이다. 이러한 제약 조건들은 강제적이고 시스템에 의해 모든 읽기와 기록 연산에 대해 자동적으로 실행되기 때문에 트로이 목마에 의한 침투를 점검하고 방지할 수 있다.

그러나, Bell-LaPadula 접근 제어 모델의 이러한 제약 조건들이 항상 올바르게 실행될지라도 보안상의 문제가 발생함이 판명되었다[11].

보안 데이터베이스 시스템은 데이터에 대한 직접적인 비밀 누출 뿐만 아니라 간접적인 비밀 누출을 통한 불법적인 정보의 흐름을 차단할 수 있어야 한다. 비밀 채널이 후자에 속하는 간접적인 비밀 누출의 형태이다. 비밀 채널은 상위 비밀 취급 인가자가 하위 비밀 취급 인가자에게 정보의 양의 많고 적음을 불문하고 간접적인 정보 누출 수단을 제공할 수 있다[7, 8].

예를 들어, 다음 [그림 2-1]과 같은 봉급 테이블이 있을 때 어트리뷰트 봉급은 기밀 데이터이다. 어트리뷰트 봉급에 대한 접근 권한이 있는 A가 접근 권한이 없는 B와 공모를 하여 0초에서 10초 사이에는 첫 번째 투플에 대한 봉급 값을 보내기로 시간 약속을 한 후 둘 다 봉급 테이블에 접근하여 충돌을 발생시킨다. 이 때 허가자 A는 충돌을 발생시키는 응용 프로그램에 시간 지연을 유도할 수 있는 트로이 목마 프로그램을 삽입하여 원하는 만큼의 수행 시간을 얻는다. 비허가자 B는 지연된 응답 시간을 측정하여 2초 후 응답을 받으면 첫 번째 투플의 봉급은 2,000,000원이라고 유추한다. 이런 방법으로 비허가자 B가 얻은 기밀 데이터가 [그림 2-2]에 나타나 있다. 즉, 허가자 A와 비허가자 B 사이에 비밀 채널을 설치한 것이다.



[그림 2-2] 비밀 채널로 인한 기밀 데이터 봉급의 유출 예

[Fig. 2-2] Example of Effluence Secure Data by Covert Channel

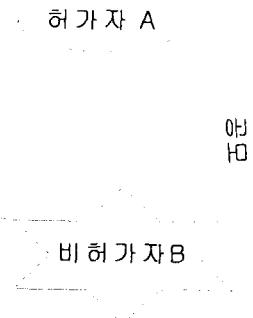
응답 시간의 지연에 따른 비밀 채널이 발생한 [그림 2-1]과 [그림 2-2]가 시간 비밀 채널의 예가 된다.

2.2 실시간 스케줄링 기법

기존의 스케줄링 알고리즘은 시스템 사용과 작업처리량을 최대화하기 위한 많은 중앙처리장치 한계와 입출력 한계의 균형을 목적으로 대부분의 기존 운영체제에서 사용되었으며, 프로세스들을 공평하게 사용하도록 설계되었다. 즉, 각각의 프로세스들은 시스템 자원의 공정한 사용을 위해 수행 작업 시간, 대기시간, 그리고 응답시간 등을 고려한 복잡한 알고리즘들을 사용한다. 그러한 이러한 기법은 종료시한의 만족률과 비밀 채널 제거를 중요시하는 보안 실시간 트랜잭션의 스케줄링에 적합하지 않다[3, 10].

테이블 : 봉급

사번	이름	직급	봉급
2001001	홍길동	사원	2,000,000
1999010	임꺽정	과장	5,000,000
2001002	장길산	대리	5,000,000
1998100	홍길동	사장	8,000,000
2001003	황진이	사원	3,000,000



[그림 2-1] 기밀 데이터 봉급을 가지는 테이블

[Fig. 2-1] Table with Secure Data

데이터와 시스템의 자원에 대한 충돌을 해결하고, 트랜잭션이 지닌 시간적인 제약조건을 만족시키고, 비밀 채널을 제거하기 위해서 보안 실시간 데이터베이스 시스템에서는 트랜잭션에 우선순위를 부과하는 기법을 사용해야 한다. 이러한 우선순위 기법으로는 가장 먼저 준비된 트랜잭션에 높은 우선순위를 할당하는 기법인 FCFS(First Come First Serve) 기법과 실시간 데이터베이스 시스템의 스케줄링에서 가장 기본적으로 이용되고 있는 기법으로 종료시한에 가장 근접한 트랜잭션에 높은 우선순위를 할당하는 EDF(Earliest Deadline First) 기법, 그리고 트랜잭션이 자신의 종료시한을 만족시키면서 수행을 얼마나 늦출 수 있는가를 나타내는 슬랙을 이용하는 LSF(Latest Slack First) 기법을 들 수 있다. 그 외 LDF(Latest Deadline First) 기법, AED(Adaptable Earliest Deadline) 기법과 종료시한과 트랜잭션이 완료되었을 때의 가치를 나타내는 중요성을 고려한 HED 기법이 있다[3].

FCFS 기법은 트랜잭션의 종료시한에 관한 정보를 사용하지 않으므로 긴급한 트랜잭션이 도착한 경우에도 먼저 대기하고 있는 트랜잭션이 앞서 처리되므로 긴급한 트랜잭션에 대해서는 대응하기가 어렵다.

EDF 기법은 종료시한 정보를 사용하므로 긴급한 트랜잭션의 처리에 용이하고, 트랜잭션들이 적당히 적재되는 경우에는 좋은 성능을 유지하지만, 트랜잭션들이 과도하게 적재되는 경우에는 종료시한에 근접하거나 종료시한을 초과하는 트랜잭션에 높은 우선순위가 할당되는 단점을 가지고 있다. 또한, FCFS 기법과 마찬가지로 비밀 채널에 대한 방지가 어렵다.

LSF 기법은 슬랙 시간으로 우선순위를 할당하는 방법이다. 슬랙 시간은 트랜잭션이 자신의 종료시한을 만족시키면서 수행을 얼마나 늦출 수 있는가를 나타내는 수치로 0 보다 크면 트랜잭션은 인터럽트 없이 진행하여 종료시한 전에 끝낼 수 있고, 0 보다 작으면 종료시한 전에 끝내는 것이 불가능하다. 이 방법의 장점은 트랜잭션 간에 충돌이 발생한 경우 높은 우선순위를 가진 트랜잭션이 슬랙 시간 내에 처리 가능하다면 낮은 우선순위를 가진 트

랜잭션을 취소하지 않고 수행함으로써 자원의 이용을 극대화할 수 있다. 그러나, 트랜잭션의 수행시간과 이를 고려한 슬랙 시간을 계산하기가 어려울 뿐만 아니라 EDF 기법에 비해 부하가 크다.

LDF 기법은 EDF 기법과는 반대되는 우선순위 할당 기법으로 종료시한에 가장 먼 트랜잭션에 높은 우선순위를 할당한다. 이 방법은 새롭게 도착한 트랜잭션이 이미 실행하고 있는 트랜잭션보다 더 늦은 종료시한을 가지는 경향이 있으므로 높은 우선순위가 할당된다. 따라서 이 방법은 EDF 기법의 약점인 과부하의 경우에 오히려 더 유리하다.

위의 스케줄링을 위한 우선순위 할당 방법은 모두 보안 데이터베이스 관점에서 볼 때, 공유 데이터에 대한 트랜잭션 충돌시 보안등급을 고려하지 못하므로 비밀 채널의 방지가 어렵다.

3. 시간 비밀 채널을 제거하는 실시간 트랜잭션 스케줄링 기법

본 장에서는 보안 데이터베이스 시스템에서 실시간 특성을 지원하기 위한 스케줄링 기법을 제안한다. 보안 실시간 데이터베이스 시스템에서 트랜잭션 처리를 위한 스케줄링 기법은 우선순위에 의한 트랜잭션 스케줄링을 요구한다. 보안 데이터베이스 시스템은 비밀 채널 생성을 방지하기 위해 보안등급이 낮은 트랜잭션을 먼저 실행해야 하므로 보안등급이 낮은 트랜잭션에 높은 우선순위를 부여하는 방법을 사용하며 실시간 데이터베이스 시스템은 트랜잭션의 종료시한 내에 완료를 목적으로 하는 시스템이기 때문에 종료시한에 따라 우선순위를 할당하는 방법을 사용한다. 그러나, 우선순위를 할당하는 두 방법은 서로 상충(trade-off) 관계이므로 충돌이 발생하면 어느 것을 중요시하여 우선순위를 할당해야 할지를 결정해야 한다. 이를 위해 본 논문에서는 보안등급을 기반으로 한 대기 큐를 사용하고, 트랜잭션의 종료시한, 중요도, 유형을 고려하여 우선순위 할당하는 스케줄링 기법을 제안한다.

3.1 트랜잭션 테이블

트랜잭션의 스케줄링을 위해서는 트랜잭션에 관한 정보가 필요하다. 이러한 트랜잭션의 정보를 유지 관리하는 트랜잭션 테이블의 구조는 [그림 3-1]과 같다.

선에 우선순위를 할당하기 위해서는 보안 특성과 실시간 특성을 모두 고려해야 한다. 그러나, 우선순위를 할당하기 위해 상충 관계인 두 특성을 동시에 반영한다는 것은 어려운 문제이다. 따라서, 본 논문에서는 보안등급의 충돌로 인한 비밀 채널의 제거를 위해 보안등급 별로

트랜잭션 ID	보안등급	종료시한	유형	중요도	마감시간	실행시간
ID	Level	Deadline	Type	Weight	ACommitTime	ExecuteTime

[그림 3-1] 트랜잭션 테이블의 구조

[Fig. 3-1] Structure of Transaction Table

트랜잭션 ID는 트랜잭션이 대기 큐로 입력될 때 부여받으며, 보안등급은 트랜잭션을 수행시킨 사용자의 보안등급을 상속받으며 트랜잭션 종료시까지 변경되지 않는 값이다. 종료시한, 유형, 중요도는 사용자가 응용 프로그램을 작성할 때 부여한 값이며, 종료시한은 트랜잭션의 시간 제약 조건이며, 유형은 트랜잭션이 하드, 펌, 소프트, 비실시간을 나타내며, 중요도는 트랜잭션의 가치를 나타내는 값이다. 실행시간은 실시간 트랜잭션의 경우에는 발생하는 종류가 한정적이므로 대부분이 시스템에 알려진 값이며, 알려지지 않은 트랜잭션의 경우에는 입력시 실행시간을 계산하여야 한다. 마감시간은 종료시한과 현재시간에 관계된 시간 값으로 트랜잭션이 완료되어야 하는 절대적인 시간이다. 타임스탬프는 해당 트랜잭션이 낮은 보안등급의 데이터를 판독하려 할 때 여러 개의 다중 버전 중 정확한 데이터 값을 판독하기 위한 값이다.

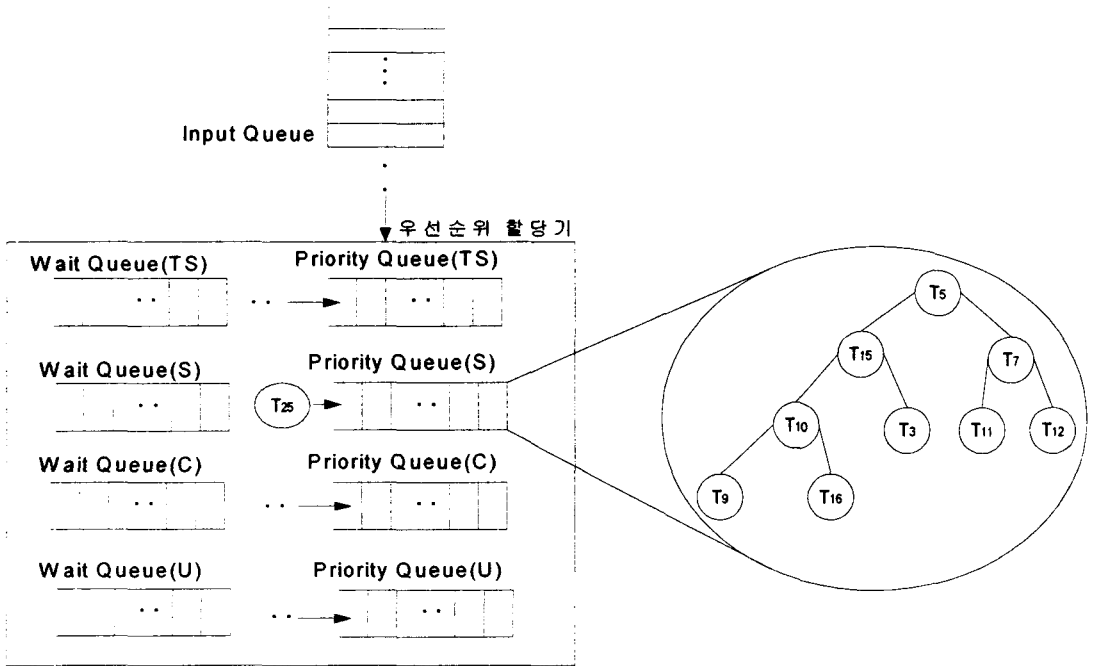
우선순위 큐를 할당한다. 이것은 트랜잭션의 우선순위를 정할 때 보안등급이 다른 트랜잭션 들끼리는 우선순위 경쟁을 하지 않도록 하여 비밀 채널 문제로 인한 종료시한에 임박한 트랜잭션을 지연시키거나 취소시키지 않도록 하는 정책이다.

다음 [그림 3-2]는 보안등급에 따라 분류한 우선순위 큐를 가진 우선순위 할당기이다. 제안한 보안 실시간 트랜잭션 관리기는 보안등급 TS, S, C, U 에 따라 4개의 분류된 우선순위 큐와 대기 큐를 가진다. 대기 큐는 입력 큐를 통해 우선순위 할당기에 삽입된 트랜잭션들 중 완료 가능성이 큰 트랜잭션들의 집합이며, 우선순위 큐는 실시간 특성에 따른 우선순위로 정렬된 힙(heap) 형태로 되어있기 때문에 트랜잭션들은 원 안의 모습처럼 정렬되어 있다.

우선순위 할당기는 대기 큐, 우선순위 큐로 구성되어 가장 먼저 실행될 트랜잭션을 결정하는 역할을 한다. 디스패처는 가장 우선순위가 높은 트랜잭션을 실행시키기 위해 동시성 제어기로 보내는 역할을 한다. 즉, 우선순위 큐의 루트 노드 트랜잭션을 동시성 제어기로 삽입한 후 삭제하는 역할을 한다.

3.2 보안등급에 기초한 우선순위 할당 기법

보안 실시간 트랜잭션 관리기에서 트랜잭



[그림 3-2] 보안등급에 기초한 우선순위 큐
 [Fig. 3-2] Priority Queue Based on Security Level

보안 실시간 데이터베이스 시스템에서 트랜잭션의 수행 시 중요한 점은 적중률을 높이기 위한 예측성의 향상이다. 즉, 실시간 트랜잭션의 스케줄링에서는 트랜잭션의 실행을 예측하고, 이를 이용하여 종료시한을 만족시키는 실시간 트랜잭션의 수를 최대화하는 것이 목적이다. 따라서 본 논문에서는 트랜잭션의 스케줄링을 위해서 트랜잭션의 제약조건을 모두 만족시킬 수 있는 트랜잭션을 선택하여 우선적으로 실행시킴으로써 트랜잭션의 종료시한과 실시간 데이터 객체의 제약조건을 모두 만족할 수 있도록 트랜잭션을 스케줄링 한다.

트랜잭션을 수행하기 전에 트랜잭션의 정보를 이용하여 해당 트랜잭션이 종료시한 내에 수행가능하고, 사용되는 데이터 객체가 트랜잭션의 완료 시점까지 유효성을 상실하지 않는지를 검사하여 수행한다. 따라서 트랜잭션 스케줄러는 트랜잭션의 수행 전에 마감시간 내의 완료 가능성을 고려하며, 트랜잭션의 성공여부를 판단하기 위하여 다음의 조건 1을 만족하는

트랜잭션을 우선적으로 선택하여 수행한다. 여기서 마감시간은 트랜잭션이 종료되어야 하는 절대적인 시간이다.

$$ACt = Ct + Dt$$

- (수식 1)

- ACt : 트랜잭션의 마감시간
- Ct : 현재 시간
- Dt : 트랜잭션의 종료시한

$$ECt = Ct + Et$$

- (수식 2)

- ECt : 트랜잭션의 완료예정시간
- Et : 트랜잭션의 예측실행시간

조건 1) 실시간 트랜잭션은 마감시간 이전에 완료되어야 한다.

$$ACt \geq ECt$$

대기 큐에 실시간 트랜잭션이 들어오면 시작시간이 알려지며, 종료시한과 예측실행시간은 트랜잭션 테이블에 있는 트랜잭션의 정보를 참조하여 알아낸다. 처음에 대기 큐에 트랜잭션이 들어오면 트랜잭션은 마감시간을 준수하지만 우선순위 큐 내에서 트랜잭션의 실행 순서를 기다리게 되면 완료예정시간은 시간의 흐름에 따라 변경된다. 그러므로, 트랜잭션 실행시에 다시 한번 수식 2에 따른 완료예정시간을 계산한 후 마감시간과 비교하여 조건 1을 만족하는지 검사한다.

조건 1을 검사하였을 때 만족하지 못하면 해당 트랜잭션은 우선순위 큐 내에서 오래 기다렸기 때문에 자신의 종료시한을 만족시키지 못한 것이므로 펌 실시간 트랜잭션인 경우에는 취소하고, 소프트 실시간 트랜잭션이나 비실시간 트랜잭션인 경우에는 상황에 따라 취소하거나 실행을 한다. 그러나, 하드 실시간 트랜잭션의 실패는 재앙적인 결과를 초래하므로 반드시 실행을 보장해야 한다. 결국 조건 1은 트랜잭션의 종료시한 만족 조건을 검사하여 실행을 결정하는 조건이며, 조건 1에 기반한 실행 여부 검사 알고리즘은 다음과 같다.

<알고리즘 3-1> 트랜잭션 수행의 타당성을 검사

입력 : 트랜잭션 T_i ,
출력 : 트랜잭션이 종료시한을 만족시키면 1, 그렇지 않으면 0

```

FeasibleTest(TransactionInfo  $T_i$ )
01: {
02:   RT_TIME   currentTime;
03:   currentTime = getTime();
04:   estimateCommitTime = currentTime +
 $T_i$ .ExecuteTime;
                                // 완료 예정 시간 계산
05:    $T_i$ .CommitTime = currentTime +
 $T_i$ .Deadline; // 마감시간 계산
06:   if (estimateCommitTime <
 $T_i$ .CommitTime)
07:     return 1; // 트랜잭션을 종료시킨 내
    
```

처리 가능

```

08:   else return 0; // 트랜잭션이 종료시한
내 처리 불가능
09: }
    
```

대기 큐에 있던 트랜잭션이 우선순위 큐로 이동이 되기 위해서는 마감시간에 여유가 있는 트랜잭션이 우선순위 큐에 포함되지 않도록 하는 임계값 α 를 고려한다. 그러므로, 대기 큐의 트랜잭션들 중에서 다음의 조건 2를 만족하는 트랜잭션들을 우선순위 큐에 포함시켜 급박한 트랜잭션들을 우선 처리하도록 한다.

조건 2) 마감시간에 임박한 실시간 트랜잭션이 먼저 실행되어야 한다.

$$ACt - Et < \alpha$$

대기 큐의 제일 앞에 있는 트랜잭션이 우선순위 큐로 이동하기 위해서는 그 트랜잭션의 마감시간과 예측실행시간의 차이를 계산하여 임계값 α 보다 큰 경우는 마감시간까지 여유가 있으므로 대기 큐에서 기다리며, 작은 경우에는 마감시간에 임박했으므로 우선순위 큐로 이동하여 실행을 준비한다. 임계값 α 는 데이터베이스 관리자가 설정하는 값으로 트랜잭션들의 수행시 임의로 설정할 수 있다. 임계값 α 의 실제 설정값은 적용되는 실시간 응용의 특성을 영향을 받으며, 트랜잭션의 도착율에 민감한 값이다. 즉, 처리해야 할 트랜잭션들이 많아질 경우에는 임계값 α 는 작아져야 하며, 그렇지 않은 경우에는 커야한다. 그러므로, α 값은 우선순위 큐 내의 트랜잭션의 수를 일정 수준으로 제어하는 역할을 한다. 조건 2는 마감시간에 여유가 있는 트랜잭션의 경우 우선순위 큐에 포함시키지 않아 우선순위 큐 내에 있는 트랜잭션들의 완료 가능성을 높이는 조건이며, 조건 2에 기반한 우선순위 큐로의 이동 알고리즘은 다음과 같다.

<알고리즘 3-2> 대기 큐의 트랜잭션을 우선순위 큐로 이동

입력 : 트랜잭션 T_i ,

출력 : void

```

MoveToPriorityQ(TransactionInfo Ti)
01: {
02:   RT_TIME   currentTime;
03:   currentTime = getTime();
04:   estimateCommitTime = currentTime +
Ti.ExecuteTime;
//완료
예정시간 계산
05:   if ((Ti.CommitTime -
estimateCommitTime) < a )
// a는 응용 특성에 따른 임계값으로
실행 지연의 한계 값
06:     Ti move to Priority Queue;
07:   else Ti wait to Wait Queue;
08: }

```

3.3 중요도와 마감시간 기반의 동적 우선순위 할당 기법

기존 보안 데이터베이스 시스템에서는 트랜잭션의 처리 시간을 빠르게 하여 평균 응답 시간을 감소시키는 목적을 가지고 있으므로 스케줄링 방법은 트랜잭션이 입력된 순서대로 처리하는 FCFS(First Come First Served) 기법을 사용한다. 한편, 기존 실시간 데이터베이스 시스템에서는 종료시한 정보를 이용하여 종료시한에 가장 근접한 트랜잭션에 가장 높게 우선순위를 부과하여 긴급한 트랜잭션을 우선적으로 처리하는 EDF기법을 사용한다. 이 기법을 사용하면 트랜잭션을 완료하는 데 남은 시간이 가장 적은 트랜잭션에 가장 높은 우선순위를 부과하기 때문에 적정 수준의 트랜잭션이 시스템에 적재되는 경우에는 종료시한 내에 처리되는 트랜잭션의 수를 최대로 할 수 있는 장점이 있다. 그러나, EDF 기법은 트랜잭션이 과도하게 적재되는 경우에 트랜잭션이 완료될 수 있는 충분한 시간이 남지 않은 트랜잭션에 높은 우선순위를 부과하기 때문에 연속적으로 트랜잭션이 종료시한을 초과한다.

또한, 실제 실시간 트랜잭션 시스템에서 발생하는 트랜잭션은 실시간 특성을 지닌 실시간 트랜잭션과 비실시간 트랜잭션이 동시에 발

생하게 된다. 즉, 시간적인 제약조건을 갖는 실시간 트랜잭션과 평균적인 응답속도를 최소화해야 하는 비실시간 트랜잭션이 동시에 발생하게 된다.

따라서 본 논문에서는 EDF 기법에 기초를 두면서 과부하 시에도 안정성이 있고, 하드 실시간 트랜잭션을 우선적으로 처리하는 것 뿐만 아니라 비실시간 트랜잭션까지 효과적으로 처리할 수 있는 트랜잭션 스케줄링 기법을 제안한다. 제안 기법은 EDF 기법과 슬랙 기법의 장점을 살려 트랜잭션이 과도하게 적재되는 경우에 발생하는 EDF 기법의 단점을 해결하도록 한다. 이를 위해 요청된 트랜잭션의 특성에 따라 도착한 트랜잭션을 유형 및 중요도와 마감시간으로 우선순위를 할당하여 하드 실시간 트랜잭션과 높은 중요도를 가지는 트랜잭션을 우선적으로 종료시한을 만족할 수 있도록 처리한다. 우선순위는 다음의 인자를 고려하여 할당한다.

우선순위 = (유형, 중요도, 마감시간)

우선순위 큐에서는 우선 트랜잭션의 유형을 검사하여 트랜잭션의 유형이 "HARD" 라면 동시성 제어기로 보내져 필요한 데이터에 대한 로크를 모두 얻은 후 바로 실행에 들어간다. 트랜잭션 유형이 HARD가 아닌 SOFT, FIRM, NON 이라면 해당 트랜잭션의 우선순위 큐 내에 올바른 위치를 찾아 삽입해야 한다. 우선은 중요도를 비교하는데 트랜잭션 T_i 의 중요도를 $W(T_i)$ 라 할 때, 새로 삽입된 트랜잭션 T_j 의 중요도가 우선순위 큐 내에 있던 T_i 의 중요도 보다 높은 경우즉, $W(T_j) > W(T_i)$ 일 때에는 트랜잭션 T_j 가 먼저 실행되도록 우선순위 큐에서 T_i 의 앞에 위치하게 한다. 두 트랜잭션 T_i, T_j 의 중요도가 동일한 경우에는 마감시간 비교를 한다. 트랜잭션 T_i 의 마감시간을 $ACt(T_i)$ 라 할 때 $ACt(T_i) < ACt(T_j)$ 이면 T_i 의 마감시간이 급박하므로 T_i 가 먼저 실행되도록 우선순위 큐에서 T_j 의 앞에 위치하게 한다. 그렇지 않으면 이미 우선순위 큐에 있던 T_j 가 T_i 앞에 위치하게 하여 먼저 실행될 수 있도록 한다. 우선순위 할당 알고리즘은 다음과 같다.

<알고리즘 3-3> 우선순위 큐에 새로 입력된 트랜잭션의 우선순위 할당

입력 : 우선순위가 부여될 트랜잭션 T_i ,
출력 : void

PriorityAssignment(TransactionInfo T_i)

```

01: {
02   TransactionInfo  $T_j$ ;
03:   if ( $T_i$ ->Type == HARD)
04:      $T_i$  move to Concurrency Controller;
05:   while(all transaction of Priority Queue)
06:   {
07:     if ( $T_i$ .Weight >  $T_j$ .Weight) {
08:       insert  $T_i$  before  $T_j$ ; exit; }
09:     else if ( $T_i$ .Weight ==  $T_j$ .Weight) {
10:       if ( $T_i$ .CommitTime <
11:          $T_j$ .CommitTime) {
12:         insert  $T_i$  before  $T_j$ ; exit; }
13:     }
14:   }
15:   repeat from step 5 to step 11 with
16:   next  $T_j$ ;
17: } // end of while
18: }
```

4. 성능 평가

본 장에서는 제안된 스케줄링 기법에 대한 성능을 평가하기 위한 모의 실험 모델과 실험 환경에 대한 내용을 설명한다. 또한, 제안된 트랜잭션 스케줄링 기법과 동시성 제어 기법에 대한 성능적 특징을 분석하고자 기존의 기법들과 성능을 비교 평가한다.

4.1 평가 환경

제안된 스케줄러는 Windows 2000 운영체제 하에서 수행하였다. 성능 평가를 위한 시스템 환경은 <표 4-1>과 같다.

<표 4-1> 시스템 환경

<Table 4-1> system environment

항목	서버	클라이언트
CPU(MHz)	pentium III 700	pentium III 950
메인메모리(MB)	785	785
하드디스크(GB)	40	40
사용 언어	visual C++	visual C++
대수	1	3

성능 평가를 위한 모의 실험 파라미터는 <표 4-2>와 같다.

<표 4-2> 모의실험 파라미터

<Table 4-2> simulation parameter

파라미터	값
Security Level	4(TS, S, C, U)
Transaction Type	4(HARD, FIRM, SOFT, NON)
Transaction Size	5 - 20
Per Operation Execution Time	3.0 ms
Weight	1 - 10
Database Size	1000 pages
Page Size	4096 bytes

성능 평가에 사용되는 파라미터는 보안등급인 Security Level을 U, C, S, TS 등급을 고려하며, 트랜잭션의 유형인 Transaction Type은 제안된 보안 실시간 데이터베이스 시스템이 모든 유형의 트랜잭션을 처리하는 시스템이므로 하드, 펌, 소프트, 비실시간의 네 가지 유형을 모두 고려하였다. 하나의 트랜잭션 발생하는 연산의 수를 나타내는 Transaction Size는 5개에서 20개 까지의 무작위로 발생하며, 하나의 연산 당 소요되는 수행 시간인 Per Operation Execution Time은 3.0ms로 고정한다. 중요도를 나타내는 Weight는 1부터 10 사

이의 값으로 동일한 비율로 발생시켰으며 모의 실험용 데이터베이스의 크기는 4MB이다.

4.2 성능 평가

트랜잭션의 스케줄링은 보안 실시간 데이터베이스 시스템에서는 기존의 실시간 데이터베이스 시스템에서 사용되는 EDF 기법을 많이 사용한다. 하지만 본 논문에서는 보안 특성과 실시간 특성을 모두 고려한 스케줄링 기법을 사용한다. 그러므로 스케줄링 기법에 대한 성능 평가는 <표 4-3>처럼 EDF, LLF+EDF 기법과 비교 평가한다. 스케줄링 기법의 성능 비교에 사용되는 주요 평가 척도는 트랜잭션이 마감 시간 내의 완료를 나타내는 성공률인 적중률이며, 하드 트랜잭션의 발생 비율에 따른 적중률과 전체 트랜잭션의 적중률을 비교한다. 제안된 시스템이 하드 트랜잭션을 우선적으로 처리하기 위한 스케줄링 기법을 사용하기 때문에 하드 트랜잭션의 발생 비율에 따라 적중률을 성능 평가한다.

<표 4-3> 성능 평가를 위한 트랜잭션 스케줄링 기법의 종류

<Table 4-3> type of scheduling method for evaluation

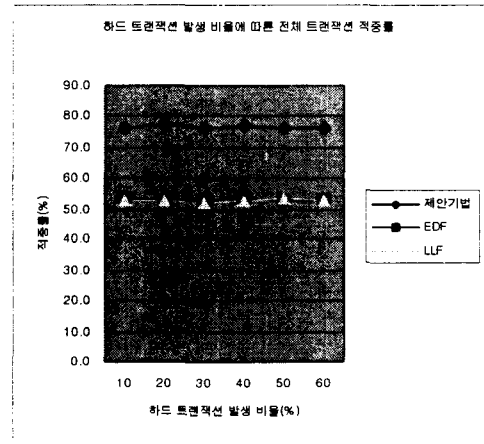
	EDF	LLF	제안기법
마감시간	○	○	○
보안등급	×	○	○
중요도	×	×	○
트랜잭션의 유형	×	×	○

마감시간의 계산은 트랜잭션 도착시간과 종료시한을 합하여 구한다. 트랜잭션의 종료시한은 실험의 공정성을 위해 모두 트랜잭션 수행 시간의 2배로 정하였다. 트랜잭션의 수행시간은 연산의 수 * 소요시간으로 계산하였다. 트랜잭션은 클라이언트에서 발생시키며, 이 때 종료시한, 보안등급, 중요도, 유형, 연산의 수를

가지고 서버로 입력된다. 클라이언트에서 트랜잭션을 발생시킬 때 네 종류의 보안등급, 유형은 동일한 비율로 발생되도록 하였다.

제안된 스케줄링 기법과 비교되는 EDF 기법은 실시간 데이터베이스 시스템에서 선호하는 스케줄링 기법인데 마감 시간만을 고려하여 스케줄링 한다. LLF 기법은 비밀 채널 방지를 위해 하위 보안등급 트랜잭션을 먼저 실행시키는 방법이며 우선순위 할당 문제로 트랜잭션 사이의 충돌이 발생한 경우에는 마감 시간을 고려하도록 하였다.

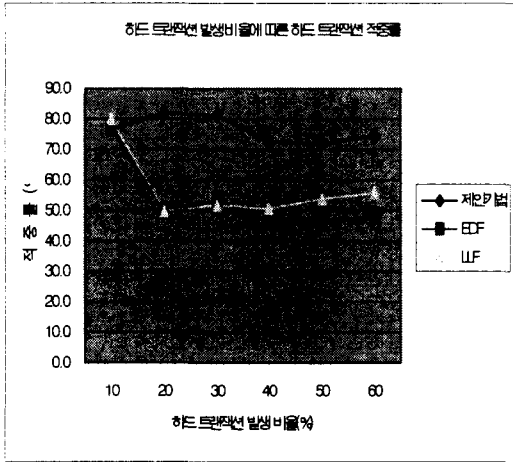
[그림 4-1]은 하드 트랜잭션 발생 비율을 10~60%로 10%씩 변경시키면서, 이에 따른 전체 트랜잭션의 적중률을 구한 것이다. 제안 기법의 성공률이 다른 두 기법보다 20~30% 높게 나타난다.



[그림 4-1] 하드 트랜잭션 발생 비율에 따른 전체 트랜잭션의 적중률

[Fig. 4-1] Hit Rate of Transactions by Hard Transaction Occurrence Rate

[그림 4-2]는 하드 트랜잭션의 발생 비율을 10~60 까지 10%씩 변경시켰을 때 하드 트랜잭션의 적중률을 구한 것이다. 보안 실시간 데이터베이스 시스템에서 하드 유형의 실시간 트랜잭션의 완료는 반드시 보장을 하여야 한다. 위의 실험과 마찬가지로 제안 기법의 성공률이 20% 이상 높게 나왔다.



[그림 4-2] 하드 트랜잭션 발생 비율에 따른 하드 트랜잭션의 성공률

[Fig. 4-2] Success Ratio of Hard Transaction by Hard Transaction Occurrence Ratio

위의 두 실험에서 다른 두 기법에 비해 제안 기법이 우수하게 나온 이유는 제안 기법에서는 트랜잭션의 마감 시간에 여유가 있으면 대기 큐에서 기다렸다 실행되므로 급박한 트랜잭션의 성공 기회가 많아졌기 때문이다. 즉, 트랜잭션 스케줄링시 α 값의 고려와 종료시한까지의 완료 가능성인 타당성을 고려함으로 인하여 제안 기법에서는 예측성이 향상되었기 때문이다. EDF와 LLF 기법의 경우 비슷한 성공률이 나온 이유는 LLF의 경우 보안등급의 종류가 네 가지 밖에 되지 않으므로 보안등급으로 변별력이 생기지 않아 결국은 EDF와 동일하게 스케줄링되기 때문이다.

5. 결론

본 논문에서는 강제적 접근 제어를 구현한 보안 데이터베이스 시스템에서 실시간 기능을 지원하는 보안 실시간 데이터베이스 시스템의 트랜잭션 스케줄링에 관한 연구를 수행하였다. 보안 실시간 데이터베이스 시스템의 트랜잭션 스케줄링시 발생하는 문제점은 실시간 특성을 유지하기 위한 우선순위 할당 정책과 데이터베이스 보안을 유지하기 위한 시간 비밀 채널 제

거 정책이 충돌을 발생시킨다는 것이다. 보안 유지의 관점에서는 보안 등급이 낮은 트랜잭션을 우선적으로 수행시킴으로 시간 비밀을 제거해야하며, 실시간의 관점에서는 종료시한에 근접한 트랜잭션을 우선적으로 수행시켜야 한다. 그러므로, 보안등급이 낮으나 종료시한은 먼 트랜잭션과 보안등급이 높고 종료시한이 급박한 트랜잭션이 충돌한 경우에 먼저 수행해야 할 트랜잭션을 선정하는 것은 어려운 문제이다.

이러한 문제를 해결하기 위해 본 논문에서는 보안 실시간 데이터베이스 시스템의 스케줄링 기법으로 보안등급에 따라 분류한 대기 큐와 트랜잭션의 유형, 중요도, 종료시한을 고려한 우선순위 큐를 유지하여 트랜잭션에 동적으로 우선순위를 부여하는 기법을 사용했다. 따라서, 제안된 기법은 시간 비밀 채널을 제거시키면서 종료시한에 근접한 트랜잭션들을 우선적으로 처리하므로 실시간 트랜잭션의 적중률을 높일 수 있었으며, 보안등급에 따라 분류한 큐로 인해 다른 보안등급을 가진 트랜잭션들 간의 불간섭 성질을 유지할 수 있었다.

본 논문에서 제안한 스케줄링 기법은 성능 평가를 통해서 기존의 기법과 비교해 20~30% 이상의 트랜잭션의 적중률을 보였다. 본 논문에서 제안된 보안 실시간 데이터베이스 시스템의 스케줄링 기법은 강제적 접근 제어를 구현하면서 시간 제약 조건을 가지는 트랜잭션을 처리해야 하는 국방 시스템 등에 사용될 수 있다. 향후 연구 방향으로서는 보안등급을 가지는 공유 데이터에 접근하는 트랜잭션들의 동시성 제어 기법과 저장 장소로 인해 발생 할 수 있는 저장장소 비밀 채널을 제거하는 회복 기법에 관한 연구가 필요하다.

참 고 문 헌

- [1] B. Purimetla, R. M. Sivasankaran, K. Ramamritham, and J. A. Stankovic, "Real-Time Databases: Issues and Applications," *Advances in Real-Time Systems*, Prentice Hall, pp.487-507, 1995.
- [2] K. Ramamritham, "Real-Time Databases," *International Journal of Distributed and parallel Databases*, pp.199-226, 1993.
- [3] 김경배, 배해영, "분류된 클래스 큐를 이용한 실시간 데이터베이스 시스템의 트랜잭션 관리기," *한국정보처리학회 논문지*, Vol.5, No.11, 1998.
- [4] B. Kao and H. Garcia-Molina, "An Overview of Real-Time Database Systems," *Real-Time Computing*, Springer-Verag, vol.127, pp.261-282, 1994.
- [5] S. Son, "Developing a Database System for Time-Critical Applications," In *Database and Expert Systems Applications* (Edited by V. Marik, J. Lazansky, R. R. Wagner), pp.313-324, Springer-Verlag, 1993.
- [6] A. R. Bakker, "Special care needed for the heart of medical information systems," *IFIP TC11 WG 11.3, 8th working conference of database security*, pp.3-10, 1994.
- [7] S. Castano, et.al., *Database Security*, Addison-Wesley, Reading, MA, 1994.
- [8] 심갑식 편저, *데이터베이스 보안*, 다성 출판사, 서울 2001
- [9] S. H. Son, R. David, and B. Thuraisingham, "An Adaptive Policy for Improved Timeliness in Secure Database Systems," *IFIP TC11 WG 11.3, working conference of database security*, 1995.
- [10] S. H. Son, R. Zimmerman, and J. Hansson, "An Adaptable Security Manager for Real-Time Transactions," 12th Euromicro Conference on Real-Time Systems, 2000.
- [11] B. Kogan and S. Jajodia, "Concurrency Control in Multilevel secure Databases Using Replicated Architecture," *Proc. ACM SIGMOD Int'l. Conf. on Management of Data*, pp.153-162, 1990.

조숙경



1990년 인하대학교 전자계산학과(이학사)

1994년 인하대학교 대학원 전자계산공학과(공학석사)

2002년 인하대학교 대학원 전자계산공학과(공학박사)

사)

2003년~현재 인천대학교 컴퓨터공학과 강의전담교수

관심분야: 데이터베이스, 실시간 데이터베이스 시스템, 데이터베이스 시스템의 보안

1996년 연세대학교 본대학원

전자공학과(공학박사)

1996년 ~ 현재 (주)케이티 서비스개발연구소 무선망설계연구실장

관심분야 : 무선망설계 시스템

개발/엔지니어링, 영상부호화

정미영



1993년 덕성여자대학교 전산학과 졸업 (이학사)

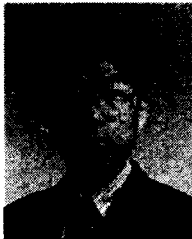
1996년 연세대학교 전산과학과 대학원 졸업 (이학석사)

1996년~ 현재 (주)케이티 서비스개발연구소 무선망설계연구실 선임보연구원

관심분야 : 무선망 설계 시스템, 지형데이터 처리 시스템, 데이터베이스시스템,

데이터마이닝

김중훈



1991년 인하대학교 전자계산공학과(공학사)

1993년 인하대학교 대학원 전자계산공학과(공학석사)

1998년 인하대학교 대학원 전자계산공학과(공학박사)

1996년 인하대학교 전자계산공학과 전임대우

1998년~현재 (주)케이지아이 대표이사

관심분야: 데이터베이스, 멀티미디어 데이터베이스 시스템, 지리 정보 시스템, SAN

배영호



1993년 포항공과대학 전자계산학과 (공학사)

1995년 포항공과대학 본대학원 전자계산학과 (공학석사)

1995년 ~ 현재 (주) 케이티 서비스개발연구소 무선망연구실 선임보연구원

관심분야: 무선망 설계 시스템 개발, 통신 시스템 성능 분석

정현민



1984년

연세대학교 전자공학과 졸업(공학사)

1986년 연세대학교 본대학원 전자공학과(공학석사)

1986년 ~ 1992년 한국통신 연구개발본부

배해영

1974년 인하대학교 응용물리학과 졸업(공학사)

1978년 연세대학교 대학원 전자계산학과 졸업
(공학석사)

1989년 숭실대학교 대학원 전자계산학과 졸업
(공학박사)

1985년 Univ. of Houston 객원 교수

1992년~1994년 인하대학교 전자계산소 소장

1982년~현재 인하대학교 전자계산공학과 교수

1999년~현재 지능형 GIS 연구센터 소장

2000년~현재 중국 중경우전대학교 대학원 명예
교수

관심분야 : 분산 데이터베이스, 공간 데이터베
이스, 지리정보시스템, 멀티미디어 데이터베
이스 등