대한의료정보학회지 제15권 제4호
J Kor Soc Med Informatics 2009;15(4):403-410

원저

의료데이터베이스에서 의료정보 보안을 위한 다중버전 관리 시뮬레이션

정현철

광주보건대학교 병원전산관리과

Simulation of a Multiversion Medical Data Management System for Medical Information Security

Hyuncheol Jeong

Dept. of Hospital Information Management, Gwangju Health College Univ.

Abstract

Objective: If medical information is integrated for management purposes, the efficiency of the system may increase. In addition, diagnostic abilities of physicians may be improved through the increased speed and accuracy of information processing. Medical databases must ensure high performance in terms of speed and reliability. In addition, access to medical information must be restricted to persons with proper authorization to ensure the privacy of patients. **Methods:** Thus, the security of medical database systems with multiversion data requires both the existing management system and security policies. **Results:** This study simulates the performance of a dynamic multiversion data management system in terms of security levels and update operations. **Conclusion:** The results show that a dynamic multiversion data management system increases disk availability more than a double version system. In addition, if the number of security levels is small, throughput will be improved because the security overhead will be low. However, frequent update operations will decrease throughput whenever versions are created at each interval. ***(Journal of Korean Society of Medical Informatics 15-4, 403-410, 2009)***

Key words: Medical Information, Database, Information Security, Multiversion, Simulation

Received for review: August 31, 2009; **Accepted for publication:** December 14, 2009

Corresponding Author: Hyuncheol Jeong, Department of Hospital Information Management, Gwangju Health College University, 683-3, Sinchang-dong, Buk-gu, Gwangju 506-701, Korea
Tel: +82-62-958-7774, **Fax:** +82-62-953-4946, **E-mail:** hcjeong@ghc.ac.kr

DOI:10.4258/jksmi.2009.15.4.403

Ⅰ. 서론

정보기술을 활용한 다양한 의료정보 처리는 새로운 분야의 개발을 유도했으며 진료 및 진단 능력에 커다란 변화를 초래하고 있다. 환자의 병력, 약제정보, 치료정보 등과 같은 다양한 의료정보가 증가함에 따라 정보기술은 대량의 검사를 신속, 정확하게 처리할 수 있게 하고 의료정보의 취급에 대한 실수를 감소시킨다. 그리고 의료정보의 분석과 처리를 수월하게 하며 객관성을 향상시킨다. 또한, 컴퓨터 네트워크를 기반으로 한 의료정보통신은 필요한 의료정보의 송수신을 가능하게 함으로써 검사 및 진료 시간, 인력과 비용을 절감시킨다. 의사와 기록부서, 회계원과 의무기록 부서, 공중보건 기관과 병원, 보험회사와 병원은 서로 협력하여 업무를 수행한다. 협력자들의 정보 교환이 필수적일지라도 데이터와 정보 자원을 전적으로 공유할 수 없다. 비인증자들에게 정보를 거절하기보다는 인증자들간의 공유정보를 보호하는 것이다. Figure 1은 양방향으로 의료정보의 내용을 검증하는 것을 나타내고 있다[1].

모든 의료분야의 활동과 밀접해진 인터넷은 보안의 중요성을 강조하게 되고 의료정보 보안은 시스템의 안정성과 신뢰성을 유지하는 필수적인 요소가 된다. 의료정보 보안은 무결성, 그리고 유용성 보장을 그 목표로 하고 있다. 특히, 여러 응용 분야에서 의료데이터베이스의 무결성, 동시성 제어, 복구 그리고 보안을 유지해야 하는 많은 노력들이 요구되고 있다. 의료정보 보안에 대한 요구 사항으로는 물리적, 논리적 의료데이터베이스의 무결성, 원소의 무결성, 의료데이터베이스내의 데이터를 접근했거나 변경한 사람을 추적할 수 있는 감사기능, 허용된 의료데이터를 접근하도록 하는 접근제어, 모든 사용자가 감사 추적과 접근이 가능한 의료데이터를 구분하도록 신분 증명을 확인하는 사용자 인증, 사용자가 허용된 범위 내에서 의료데이터베이스를 접근할 수 있는 가용성을 들 수가 있다. 의료데이터베이스는 운영 체제와 밀접한 연관성을 갖고 있고 정부, 기업, 소규모 집단에서 정보를 처리하는데 큰 비중을 차지한다. 또한, 의료데이터베이스는 시스템 소프트웨어이면서 긴밀한 정보를 유지하고 있고 이런 중요한 정보는 법적으로 보호되어야 한다. 그렇기 때문에 의료데이터베이스에 저장된 데이터로의 불법적인 접근, 고의적인 파괴, 변경, 그리고 우발적인 사고로 인한 데이터의 일관성 위배로부터 데이터 혹은 데이터베이스를 보호하고자 하는 의료데이터베이스 보안은 매우 중요하다. 의료데이터베이스로의 접근을 제어하는 정책은 임의의 접근 제어와 강제적 접근 제어를 들 수 있다. 강제적 접근 제어는 임의 접근 제어와는 달리 주체와 객체에게 각각의 보안등급을 할당하여 주체가 객체로의 접근을 제한한다. 강제적 접근 제어 정책은 BLP[2-4] 모델을 따른다. 즉, 주체의 보안 등급보다 높은 객체의 판독은 허용하지 않으며 기록은 허용한다. 이러한 제약 조건을 따른다면 높은 보안등급의 객체에서 낮은 보안 등급의 객체로 직접적으로 정보가 흐르는 것을 차단한다. 그러므로, 강

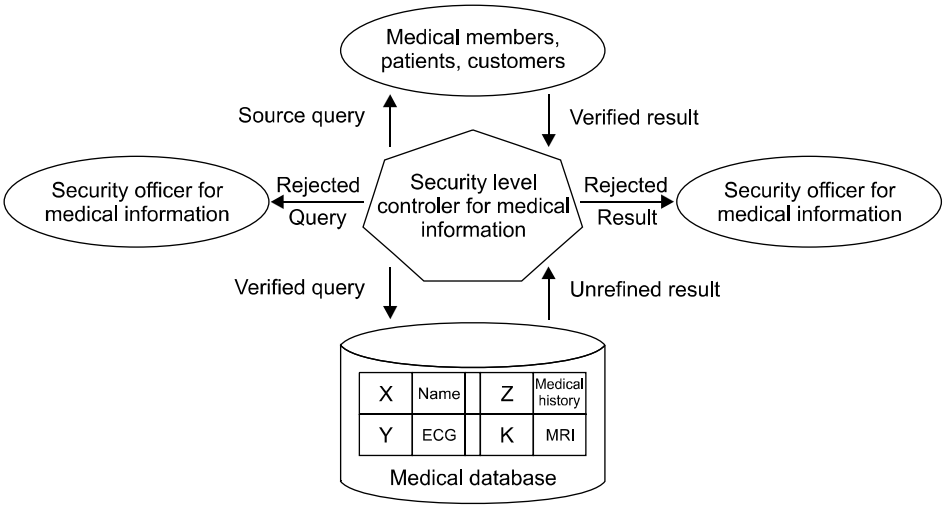

Figure 1. Bidirective authentication of medical information contents

제적 접근 제어는 임의 접근 제어에서 발생될 수 있는 트로이 목마 문제를 완화시킨다.

따라서, 본 연구에서는 강제적 접근 제어 정책에 따라 보안 성질을 고려하여 동적으로 다중버전을 제어하는 기법[5](이하 MLS/DMVC)에 대해서 시뮬레이션을 통하여 성능을 분석하였다. 본 연구의 구성은 II장에서는 한 데이터 항목이 두개 이상의 버전을 갖는 버전 관리에 대해 언급하고 III장에서는 보안 환경에서 동적으로 다중버전을 관리하는 알고리즘의 개념을 서술한다. 그리고 시뮬레이션을 통하여 기존의 방법들과 비교 분석하고 IV장에서는 고찰을 기술한다.

II. 재료 및 방법

의료데이터베이스 시스템에서는 모든 환자의 정보를 공동으로 등록 및 조회하여 데이터를 수집하고 각 개인의 사생활을 보장하기 위해 이러한 의료데이터를 보호하는 것이 아주 중요한 요소가 된다. 이 장에서는 관련 연구로써 의료데이터베이스 시스템이 각 데이터 항목의 버전에 대해 보안등급을 고려한 트랜잭션들을 스케쥴링하는 기법들과 장단점을 살펴본다.

Keefe[6]가 제안한 이 기법은 도착하는 트랜잭션들에게 각각의 순서스탬프를 할당한다. 보안등급의 관계가 $L(P)<L(T)$일 경우 트랜잭션 T의 순서스탬프가 수행중인 트랜잭션 P의 순서스탬프보다 작고 보안등급의 관계가 $L(Q)>L(T)$일 때 수행중인 트랜잭션 Q보다 크게 보장되도록 트랜잭션 T의 순서스탬프는 결정되어 진다. 여기서, L(T)는 T의 보안등급을 표시한다. 이 기법은 하위등급에서 수행중인 모든 트랜잭션들에게 할당된 것보다 더 작은 순서스탬프를 상위 트랜잭션들이 할당받는다. 그리고 그 순서스탬프대로 트랜잭션들을 수행한다. 이렇게 상위 등급 트랜잭션들에게 더 작은 타임스탬프를 할당하는 이유는 하위 등급에서 수행중인 트랜잭션들과의 충돌을 방지하려 하기 때문이다. 두 트랜잭션들 사이의 충돌은 상위 등급 트랜잭션이 이미 판독한 버전에 대해 상위 등급 트랜잭션 보다 이전의 타임스탬프를 갖는 하위 등급의 트랜잭션이 기록할 때 발생한다. 따라서, 상위 등급 트랜잭션의 판독 연산은 무효화된다. 이것은 전형적인 이전 시점 방법에 해당된다. 결국, 상위등급 트랜잭션은 자신 보다 큰 순서스탬프를 갖는 하위등급 트랜잭션이 생성한 것보다 오래된 버전을 판독한다. 상위등급 트랜잭션이 판독하고자 하는 버전의 최근성은 하위등급에서 수행중인 트랜잭션에 달려 트랜잭특히, 하위등급에서 수행중인 트랜잭션이 장기 트랜잭션이면 상위등급 트랜잭션은 상당히 오래된 버전을 판독하게 된다. Jajodia[7]가 제안한 이 기법에서는 트랜잭션이 시스템에 도착하는 순서대로 트랜잭션에게 타임스탬프를 할당해서 타임스탬프 순서대로 트랜잭션을 수행한다. 보안등급 관계가 $L(T)>L(P)$이고 타임스탬프의 관계가 $ts(T)>ts(P)$일 경우에 하위등급에 있는 P의 모든 트랜잭션들이 완료될 때까지 트랜잭션 T의 완료는 연기된다. 만일, $L(T)>L(Q)$이고 $ts(T)>ts(Q)$일 때, T가 객체 x의 버전 x_i를 판독한 후에 x의 새로운 버전 x_j를 트랜잭션 Q가 생성한다면, 트랜잭션들은 타임스탬프 순서대로 수행되기 때문에 T의 $r(x_j)$는 재수행 되어야 한다. 하위등급에 장기 트랜잭션 T^L이 있다면 $ts(T^H)>ts(T^L)$인 상위등급 트랜잭션 T^H의 완료는 오랜 시간 동안 연기될 수 있다. 최근에 Atluri[8]는 트랜잭션들이 다른 최근성을 선택할 수 있도록 하는 기법을 제안했다. 즉, 각 트랜잭션들이 원하는 최근성을 갖는 데이터를 판독한다. 그렇지만 이 기법이 갖는 문제는 다른 등급에 있는 데이터를 접근하는 트랜잭션들이 연속적으로 재수행 되는 것과 몇몇 트랜잭션들이 잠재적으로 기근 문제를 갖는 것이다.

Pal[9]는 장기 트랜잭션을 지원하기 위하여 보안 환경에서 이중버전을 갖는 데이터베이스에 대한 잠금 프로토콜을 제안했다. 이 프로토콜은 각 객체들에 대해서 두 개의 버전 즉, 옛 버전과 현재 버전을 유지한다. 상위등급에 있는 트랜잭션은 하위등급에 있는 옛 버전을 판독한다. 객체와 동일등급에 있는 트랜잭션은 현재 버전을 접근한다. 버전간격이 변화된 후에 현재 버전은 옛 버전으로 복사된다. 이 프로토콜이 갖는 제약사항은 한 트랜잭션의 모든 하향판독 연산들과 갱신 트랜잭션들은 한 버전간격에서 수행되어야 한다. 따라서, 장기 트랜잭션의 수행 시간대에 있는 트랜잭션들은 장기 트랜잭션이 판독한 동일한 버전들을 판독해야 한다. 이 프로토콜은 다른 다중버전 프로토콜이 갖는 기억 장소와 접근 부담을 해결하지만 다음과 같은 잠재적인 문제를 갖는다. 첫째는 버전간격이

길게 되면 될수록, 하향 판독하는 연산들은 더욱 더 오래된 버전을 판독하게 된다. 둘째로는 버전간격이 변화할 때, 갱신 트랜잭션들의 숫자가 변화할 때, 한 트랜잭션의 하향판독 연산들의 숫자가 증가할 경우, 트랜잭션들이 철회되는 비율은 증가된다.

III. 결과

생체 신호 처리, 혈액과 대소변 검사, 진료수가 청구, 의료영상 정보 등과 같은 환자에 대한 의료정보는 디지털화되어 의료데이터베이스에 저장되고 필요에 따라서 입원실, 진료실, 그 외 의료요에련 기관에서 접근되어 직접 활용될 수 있다. 이 장에서는 보안성이 고려된 의료데이터베이스에서 동적 다중버전 관리의 알고리즘을 서술하고 시뮬레이션을 통하여 기존 방법들과 비교분석한다.

1. 동적 다중버전 관리

다중버전은 버전의 생성을 적절히 조정하고 제거함으로써 동적으로 유지할 수 있다. 트랜잭션이 제출될 때마다 보안등급 관리자(이하 SLM), 트랜잭션 관리자(이하 TM), 레벨 스케쥴러(이하 LSCH), 의료데이터 관리자(이하 MDM)에 대한 알고리즘이 호출된다. MDM은 새로운 버전이 생성된 후에 가장 작은 타임스탬프를 조정해주고 사용되지 않는 버전을 제거한다. SLM에서는 도착한 트랜잭션들을 전체적 리스트에 유지하며 트랜잭션들을 보안 등급별로 분류하여 해당되는 등급의 TM에게 제출한다. 또한, 이 트랜잭션들에게 타임스탬프 생성 규칙에 따라 유일한 타임스탬프를 부여한다. TM에서는 트랜잭션들을 자신의 등급에 해당하는 기록연산과 하위등급에 있는 버전을 판독하는 연산으로 분해하고 이 연산들을 각각 해당 등급의 LSCH에게 제출한다. LSCH에서는 TM에서 받은 각 연산에 대해서 하향 판독 연산(이하 rdop)의 경우 T_j의 최초 연산이면 이것의 타임스템프(이하 rdts(T_j))를 결정하며 타임스탬프 순서 규칙에 따라 연산들의 등급을 구분하지 않고 스케쥴링하며 단순 연산인 (r(x))를 버전 연산인 (r(x_j))로 번역하여 MDM으로 제출한다. MDM에서는 LSCH에서 받은 연산들을 수행하여 버전을 생성할 것인지의 여부를 결정한다. 새로운 버전을 생성한 후에는 필요한 변수를 초기화한다. 또한, 상위등급 트랜잭션이 사용한 버전의 집합(이하 hvs)과 현재 버전(이하 cv)에 새롭게 생성된 버전을 반영하며 등급 i에서 아직 버전화 되지 않은 갱신의 가장 빠른 타임스탬프(이하 ets(i))를 조정한다. 드물게 일어나는 갱신을 반영하기 위한 고정된 간격의 버전 생성 시점(이하 VCP)에서는 이전 VCP 이후에 갱신 연산이 수행된 모든 데이터 항목에 대해서 새로운 버전이 생성된다. 의료데이터 항목 x에 대해 대기 카운터(이하 wait_cnt(x))는 처음 결정되어진 x에 대한 타임 간격(이하 TI(x))내에서 다른 갱신이 발생하지 않을 경우 VCP 이전에 버전을 생성하기 위해 기다릴 수 있는 TI(x)의 횟수를 나타내는 것으로 wait_cnt(x)의 초기값은 새로운 버전을 생성하기 위하여 시스템에서 정한 갱신 횟수(이하 UCS)와 같고, 갱신이 발생하거나 TI(x)가 한 번 만료될 때마다 1씩 감소한다. wait_cnt(x)가 0이 되면 새로운 버전을 생성한다. MDM에서 유지하는 갱신 리스트(이하 UL)에는 갱신 상태와 TI(x), 그리고 VCP가 시간에 따라 유지된다. 새로운 버전은 Ucnt(x)가 UCS를 만족하거나, wait_cnt(x)=0이 되거나, VCP 시점에 도달하는 경우에 생성된다. 새로운 버전이 생성되면 사용되는 변수들은 초기화되고 상위등급 트랜잭션이 사용하는 버전의 집합인 hvs에는 버전이 추가된다. 만약, 버전화 하려는 갱신 값이 없으면 T_j는 UL에서 제거된다. 동일등급의 트랜잭션들이 사용하는 cv는 갱신 카운터가 만족되어 새로운 버전이 생성될 때마다 상위등급의 rdop들을 위하여 주기적으로 hvs으로 삽입되기 때문에 많은 옛 버전들이 존재하게 된다. 따라서, 상위 등급 트랜잭션의 rdop가 더 이상 접근하지 않은채 hvs에 있는 옛 버전들을 제거하게 되면 디스크 공간은 절약된다. 따라서, 불필요한 버전 제거(이하 DUV)가 이루어진다. MDM은 사용되지 않는 버전을 제거하기 위해 해당 버전에 대한 하향판독 연산자의 타임스탬프를 SLM에게 전송하여 확인을 한다. SLM에는 수행중인 트랜잭션들의 리스트가 유지되기 때문에 수행을 마친 트랜잭션을 확인할 수 있고 LSCH는 단순 연산을 버전 연산으로 번역하기 때문에 어떤 트랜잭션이 어떤 버전을 사용하고 있는지를 안다. 불필요한 버전 관리를 효율적

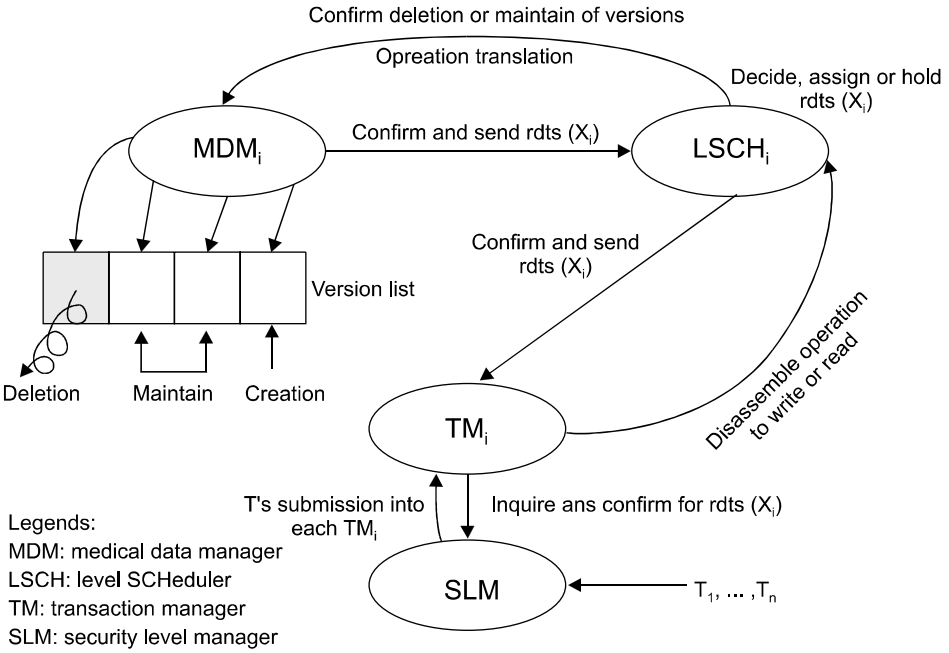

Figure 2. Diagram of algorithm

으로 하기 위하여 DUV를 위한 윈도우를 사용한다. Figure 2는 알고리즘의 간단한 도식화를 나타낸다.

2. 시뮬레이션

이 절에서는 시뮬레이션을 위해 필요한 가정과 시뮬레이션 모형 그리고 각 평가 변수를 서술하고 보안등급에 따른 시뮬레이션 그래프를 보인다.

(1) 시뮬레이션 환경

보안을 고려한 동시성 제어에서는 성능에 영향을 미치는 중요한 요소로 보안 규약을 고려할 수 있다. 그래서, 서로 다른 보안등급을 갖는 트랜잭션들간에 보안성을 고려한 동시성 제어 때문에 보안은 시스템의 성능에 부담으로 작용한다. 비밀경로 문제가 효과적으로 해결되고 성능이 좋은 알고리즘이 요구된다. 기존 연구들이 모두 비밀경로의 형성 없이 트랜잭션들이 스케쥴링 되도록 하는 같은 목표를 가지고 있다. 하지만, 이들은 각자 접근하는 연구 방식에 차이가 있음을 알 수 있다. 따라서, 동일한 실험을 수행하기 위하여 다음과 같은 환경이 필요하다.

1) 가정

기존 연구들이 서로 같지 않은 알고리즘과 시스템 모형을 갖기 때문에 성능을 평가하기 위해서는 몇 가지 가정이 필요하다. Graubart[10]는 데이터베이스를 이루는 데이터의 구성 관점에 따라서 단일화된 데이터베이스 관리 시스템(이하 DBMS)와 중복된 DBMS로 분류한다. 단일화된 DBMS는 같은 보안등급을 갖는 데이터로 데이터베이스가 구성된 것이고 중복된 DBMS는 자신과 같은 데이터 보안등급의 데이터와 하위 보안등급의 데이터로 데이터베이스가 구성된 것이다. 본 연구에서는 보안성을 높이기 위하여 기존의 연구에서 대부분 사용하는 단일화된 DBMS로 한다. 그리고 데이터를 갱신하는데 있어서 Keefe[6]에서는 트랜잭션과 보안등급이 같은 데이터를 갱신하는 경우와 상위 보안등급을 갖는 데이터를 갱신하는 경우로 분류하고 있다. 알고리즘과 실험환경을 단순화하기 위하여 본 연구에서는 동일한 보안등급을 갖는 경우에 갱신이 가능하도록 한다. 또 철회된 트랜잭션은 트랜잭션 관리자에 제출되어 즉시 수행된다고 한다. 상위 보안등급의 트랜잭션은 비밀경로 문제를 해결하기 위하여 자주 철회될 수 있는데 이는 성능에 좋지 않은 영향을 미치게 된다.

2) 시뮬레이션 모형

시뮬레이션에 사용할 트랜잭션의 실험모형은 Figure 3과 같다. 보안 트랜잭션 생성자(이하 STG)는 의료데이터베이스로 접근하는 사용자들이 프로세스를 합당하게 수행하도록 보안 정책 관리자(이하 SPM)에 근거하여 트랜잭션을 생성시킨다. SPM은 생성된 트랜잭션에 대하여 보안 정책의 위반 여부를 관리한다. 트랜잭션이 합당한 연산으로 구성되어 있으면 트랜잭션 관리자 큐(이하 TMQ)로 보낸다. TM은 TMQ에서 연산을 받아 보안등급 스케쥴러 큐(이하 LSHCQ)로 보낸다. 그리고 LSCH와 MDM에서 메시지를 받아 결과를 사용자에게 전송한다. LSCH는 LSCHQ에서 연산을 받아 처리한다. 현재 수행중인 트랜잭션의 연산을 받으면 종료 가능 여부를 고려하여 대기 큐(이하 WQ)로 보내던지 종료한다. 즉시 실행이 가능하면 의료 데이터 관리 큐(이하 MDMQ)로 연산을 전송한다. MDM은 MDMQ에서 연산을 가져와 연산의 보안등급과 동일한 데이터를 접근하게 한다. 접근이 완료되면 MDM은 TM에게 연산의 완료에 대한 응답 메시지를 보낸다.

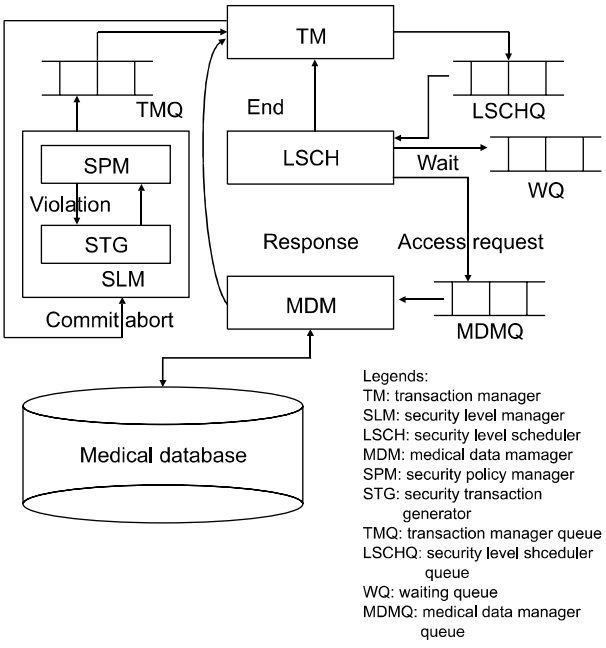

Figure 3. Simulation model

3) 매개 변수

시뮬레이션에 사용된 대표적인 매개변수는 관련연구[11-13]에서와 비슷하며 다음과 같다. 보안등급의 수(sl_num)는 여러 보안등급을 갖는 트랜잭션들이 동시에 수행되기 때문에 동시성 제어를 위하여 사용되는 변수로써 여러 보안등급을 갖는 트랜잭션들 사이에 충돌 현상을 볼 수 있다. 보안등급은 상수로 정해진다. 의료데이터베이스의 크기(db_size)는 대부분의 성능 연구에서 1,000으로 정해진다. 보안 트랜잭션이 생성되는 수(stg_num)가 많으면 동시에 수행되는 트랜잭션의 수가 많아지고 연산들의 충돌이 증가된다. 이는 트랜잭션의 동시성을 제어하기 위한 변수이다. 버전의 개수(ver_num)는 트랜잭션의 갱신 연산에 의해서 생성된 버전의 수이다. 시뮬레이션 시간(simul_time)는 트랜잭션들이 충분히 수행될 수 있도록 3,000초로 설정하였다. 트랜잭션의 크기(tran_size)는 트랜잭션을 구성하고 있는 연산자들의 개수를 나타낸다. 갱신 연산 비율(up_rat)은 갱신 연산이 수행될 때 트랜잭션들간에 충돌을 발생시킴으로 성능에 중요한 의미를 갖는다. 갱신 연산의 비율은 충돌 현상과 비례한다.

4) 평가 변수

Keefe[6]와 Pal[9] 그리고 MLS/DMVC의 성능을 비교평가하기 위하여 시뮬레이션을 통하여 측정한 대표적인 평가변수는 다음과 같다. 하향판독 버전 응답 시간(rdts_res)은 하향 판독하고자 하는 버전을 결정하는 시간이며 이때 보안등급이 같은 경우 디스크 점유율을 비교한다. 그리고 보안등급간 처리 비율(sl_rat)은 보안등급 수에 따른 트랜잭션들간의 처리율을 측정한다. 다중버전 유지 비율(mver_rat)은 버전의 생성 간격에 따라서 버전을 유지한다. 이때 갱신 연산에 의한 변화를 비교한다.

시뮬레이션 프로그램은 TC Win 4.5 C++언어를 사용하여 윈도우 XP로 운영되는 IBM PC 시스템에서 구현하였다.

(2) 성능 평가

1) 의료정보 보안등급이 같은 경우

이 절에서는 보안등급이 동일한 경우에 각 기법에

대하여 디스크의 점유율을 살펴보았다. Keefe[6]는 다중버전 그리고 Pal[9]은 이중버전을 유지한다. Figure 4에서 처럼 Keefe[6]의 경우는 트랜잭션들에 의한 디스크 점유율이 꾸준히 상승됨을 보여 주지만 Pal[9]은 버전 생성 간격이 정적으로 일정하여 항상 두 개의 버전을 유지함으로 디스크 점유율은 거의 일정하다. 그러나, 다중버전을 동적으로 관리하는 MLS/ DMVC의 경우는 rdts_res에 의해서 더 이상 사용되지 않은 버전을 제거하는 시점에서는 디스크 점유율이 다소 하강했다가 동적인 버전 간격에 의해서 버전이 생성되면 증가하는 상태를 유지함을 알 수가 있다.

2) 의료정보의 보안등급이 다른 경우

트랜잭션의 크기가 일정한 상태에서 보안등급 수가 적을수록 처리율이 향상되다가 보안등급 수가 많아지면 처리율은 유사하게 되는 경향을 보인다. MLS/ DMVC는 하향 판독 버전을 결정하는 시기가 Keefe[6]와 Pal[9]에 비해서 트랜잭션의 최초 연산이 수행되는 시점이며 재수행이 발생하지 않고 동적으로 버전을 생성하며 유지하기 때문에 상위 등급의 트랜잭션이 철회될 가능성이 적다. Figure 5는 보안등급 수의 변화에 따른 처리율을 보인다. 일반적으로 보안등급은 4등급으로 구분되어진다. 그런데 보안등급의 수가 한 등급 즉, 보안등급 수가 1일 때는 보안등급이 모두 동일하다는 의미와 같다. 그래서 이 절에서는 보안등급 수의 변화에 따른 각 기법의 처리율을 살펴보기 위하여 1일 때는 처리율이 0이 되도록 하였고 보안등급의 수는 일반적인 보안등급 수 보다 더 세분화됨을 고려하여 10등급으로 정하였다. 보안등급의 수는 간략화되거나 더 세분화 될 수도 있다.

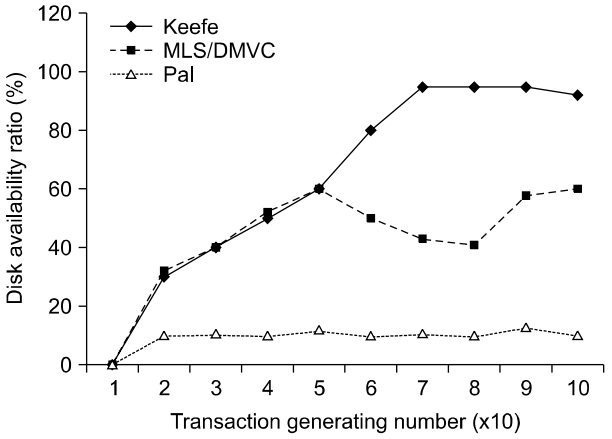

Figure 4. Disk availabilities

3) 갱신 연산에 의한 변화

이 절에서는 갱신 연산에 대한 처리율을 분석하기 위하여 db_size는 1,000으로 설정하고 갱신 연산자의 수는 트랜잭션의 크기에 대하여 25%로 설정하여 처리율을 살펴보았다. 주로 갱신 연산의 수가 높을수록 트랜잭션들간의 충돌이 발생할 가능성이 높고 버전 생성 간격에 따라서 디스크를 접근하는 시간이 요구되므로 시스템에 오버헤드로 작용하여 성능을 저하시킨다. Figure 6은 갱신연산 변화에 따른 처리율을 보인다. MLS/DMVC는 버전 생성 간격이 동적이므로 버

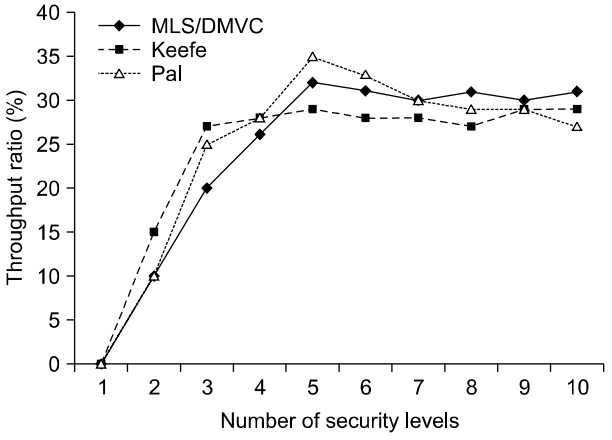

Figure 5. Throughput of different security levels

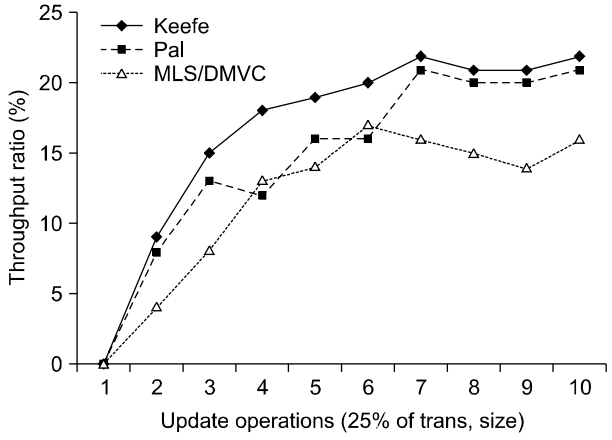

Figure 6. Throughput of update operations variation

전 생성이 드문 부분에서는 처리율이 향상되지만 버전 생성이 빈번한 부분에서는 연산자들간의 충돌현상이 발생하므로 처리율이 낮아짐을 보였다.

Ⅳ. 고찰

환자에 대한 의료정보는 인증 받은 의료관계자가 부분적으로 접근할 수 있도록 허용되어 의료정보가 선택적으로 보호될 필요성이 있다. 현재 의료 관련 기관에서 환자의 프라이버시에 대한 정보보호가 권고되고 있지만 실제로는 많은 상황이 위험에 노출되어 있다. 의료데이터베이스의 보안에서 트랜잭션의 직렬성과 보안성[14] 그리고 복구 관리[15]를 보장함으로써 의료데이터베이스 일관성과 신뢰성을 유지시킨다. 다중버전 방법은 너무 많은 버전의 생성으로 디스크 공간의 부담을 초래한다. 본 연구에서는 이러한 문제를 완화시키기 위해 동적으로 다중버전을 제어 방법에 대해서 기존의 방법과 비교분석하였다. 기존의 다중버전 방법[6-8]에서는 시스템으로 제출되는 트랜잭션이 특정한 시점만을 요구한다면 트랜잭션의 재수행 문제, 기근 문제 그리고 아주 오래된 버전을 판독하는 문제를 갖게 된다. 또한, 다중버전 유지는 많은 디스크 공간 문제를 갖는다. 이중버전 방법[9]은 버전 생성 간격과 장기 트랜잭션의 성공적인 수행 시간이 종속적이다. 보안 환경에서 동적으로 다중 버전을 제어하는 방법에서는 트랜잭션의 최초 연산이 수행될 때, 적절한 버전을 판독함으로써 오래된 버전 판독 문제, 재수행 문제, 기근 문제를 해결하고 버전 생성 간격을 조정하고 불필요한 버전을 제거하여 버전의 갯수를 동적으로 유지함으로써 디스크 공간 문제를 해결한다. 본 연구에서는 의료정보의 보안등급이 같을 경우 디스크 점유율과 보안등급이 다를 경우 처리율 그리고 갱신 연산 변화에 의한 처리율을 시뮬레이션하여 기존의 연구와 성능을 비교하였다. 향후 연구로는 각 보안등급별로 각각 구축된 시스템은 성능에 상당한 부담으로 작용함으로 보다 단순화된 시스템에서 보안성질을 기반으로 한 트랜잭션들의 직렬성을 고려하여 다중 버전에 대한 관리의 성능 평가가 이뤄져야 한다. 특히, 이질형인 환경에서는 지금까지의 동질적인 환경과는 달리 각 지역의 자치성에 따라서 버전과 보안 관리 기법이 다르기 때문에 이에 대한 연구와 성능 향상이 고려되어야 할 필요성이 있다.

참고문헌

1. Jeong HC. The security of medical information system. KISS 1998;16(12):141-145.
2. Bell DE, LaPadula LJ. Secure computer systems: unified exposition and multics interpretation. Technical Report MTR-2997 Mitre Corp;1976.
3. Castano S. Database security. Addison-Wesley;1994. pp.82-96.
4. Pfleeger CP. Security in computing. Prentice Hall; 1989. pp.249-250.
5. Jeong HC. Dynamic multiversion control in multilevel security environments. KIPS 1997;3(4):123-130.
6. Keefe TF, Tsai WT. Multiversion concurrency control for multilevel secure database systems. IEEE:Security and Privacy;1990. pp.369-383.
7. Jajodia S, Atluri V. Alternative correctness criteria for concurrent execution of transactions in multilevel secure databases. IEEE:Security and Privacy;1992.
8. Atluri V, Bertino E, Jajodia S. Providing different degrees of recency options to transactions in multilevel secure databases. IFIP:WG 11.3 Database Security; 1995. pp.199-221.
9. Pal S. A locking protocol for multilevel secure databases providing support for long transactions. IFIP:WG 11.3 Database Security;1995. pp.119-121.
10. Graubart R. A comparison of three secure DBMS architectures. IFIP:WG 11.3 Database Security; 1989. pp.130-135.
11. Care M, Stonebraker M. The performance of concurrency control algorithm for database management systems. IEEE:10th VLDB;1984. pp.107-118.
12. Agrawel D, Abbadi A, Lang A. Performance characteristics of protocols with ordered shared locks. IEEE: 7th Data Engineering;1991. pp.592-601.
13. Choi YG. The performance of concurrency control in multilevel secure database management system. MS thesis:Kaist;1995.
14. Jeong HC, Lim CG. Transaction serializability with security in heterogeneous medical database systems. J of KOSMI 1999;5(3):109-118.
15. Jeong HC. Multilevle secure recovery management of medical databases in hospital information system. J of KOSMI 2000;6(2):17-25.