

트랜잭션의 중요도와 데이터의 유효성을 고려한 실시간 이동 트랜잭션 관리자의 설계

조숙경*, 안준순*, 김경배**, 이순조***, 배해영*

*인하대학교 전자계산공학과

**전자통신연구원 컴퓨터·소프트웨어 연구소

***서원대학교 컴퓨터정보통신 공학부

g9741139@inhavision.inha.ac.kr

Design of Real-Time Mobile Transaction Manager considering Value of Transactions and Validity of Data

Sook-Kyoung Cho*, Jun-Soon Ahn*, Gyoung-Bae Kim**, Soon-Jo Lee***, Hae-Young Bae*

*Dept of Computer Science & Engineering, Inha University

**Computer & Software Technology Laboratory, ETRI

***School of Computer, Information & Communication, Seowon University

요약

본 논문에서는 이동 컴퓨팅 환경에서 발생하는 실시간 이동 트랜잭션을 효율적으로 처리하기 위한 트랜잭션 관리자를 제안한다. 제안된 트랜잭션 관리자는 기존의 종료시한만 고려하던 방법과는 달리 이동 호스트 때문에 발생하는 실시간 데이터의 유효성도 함께 고려하여 트랜잭션의 실행여부를 판단한다. 그 후, 트랜잭션의 중요도와 종료시한을 고려한 우선순위 큐에 최적의 실행 원도우를 생성하여 스케줄링하고, 실행될 트랜잭션을 선택할 때는 이동 호스트와의 접속 단절을 고려한다. 따라서, 제안된 기법은 실시간 데이터의 유효성 조건 위반에 따른 트랜잭션의 철회 후 재시작을 감소시키며, 종료시한을 만족시키는 실시간 이동 트랜잭션의 중요도의 합을 최대로 하는 장점을 가진다. 이 기법은 이동 통신, 응급 재난 구조 시스템 등에서 발생하는 실시간 이동 트랜잭션의 관리자로 적합하다.

1. 서론

최근 들어 휴대용 컴퓨터의 대중화와 무선 매체들의 급속한 발전으로 인해 사용자가 네트워크의 접속을 유지하면서 원하는 장소로 자유로운 이동을 할 수 있는 이동 컴퓨팅(mobile computing)[1,2]이 가능하게 되었다. 이로 인해 사용자들이 고정망에서뿐만 아니라 이동 컴퓨팅 환경에서도 고정망 수준의 실시간 처리 능력을 요구하게 됨으로써 이동 컴퓨팅 환경에서의 실시간 데이터베이스 시스템에 대한 요구가 크게 증대되었다[3].

이동 컴퓨팅 환경을 위한 실시간 데이터베이스 시스템에서 발생하는 실시간 이동 트랜잭션(real-time mobile transaction)은 이동 트랜잭션(mobile transaction)[4]과 실시간 트랜잭션(real-time transaction)[5]이 결합된 형태이다. 이동 트랜잭션은 트랜잭션 결과의 정확성이 위치에 영향을 받으며, 이것은 이동 트랜잭션의 위치 데이터가 실시간 데이터임을 의미한다. 이러한 실시간 데이터는 동적으로 끊임없이 변화하므로 트랜잭션의 종료 시점까지 유효해야 한다. 이것을 실시간 데이터의 유효성(validity) 이라 한다. 실시간 트랜잭션은 종료시한(deadline)이라는 시간 제약 조건을 가지며 이 조건은 트랜잭션 결과의 가치가 정확성뿐만 아니라 시간에도 영향을 받는 조건이다.

실시간 이동 트랜잭션은 종료시한과 더불어 데이터의 유효성을 만족시켜야만 정확한 결과를 도출할 수 있다. 그러나, 이동 호스트의 이동성과 좁은 대역폭의 무선통신 채널 사용으로 인해 이동 호스트와 고정망과의 통신 시간 지연으로 전체적인 트랜잭션 처리 시간이 길어지게 되어 트랜잭션의 종료시한을 만족시키지 못한다는 문제점이 발생한다[6]. 그러므로, 본 논문은 이러한 문제점을 해결하고 실시간 이동 트랜잭션의 종료시한과 실시간 데이터의 유효성을 보장하기 위한 실시간 이동 트랜잭션 관리자를 제안한다.

제안된 트랜잭션 관리자는 실행여부 검사기, 우선순위 할당기, 스케줄러, 디스패처(dispatcher)로 구성되어 있다. 실행여부 검사기는 트랜잭션이 종료시한 내에 수행 가능하고, 실시간 데이터가 완료(commit) 시점까지 유효성을 상실하지 않는지를 검사하여 실행 여부를 판단한다. 실행이 결정된 트랜잭션은 우선순위 할당기로 보내지며, 우선순위 할당기에서는 실시간 트랜잭션일 경우에는 EDF(Earliest Deadline First) 기법[7]과 트랜잭션의 중요도(value)를 기반으로 한 우선순위 큐에 트랜잭션을 삽입하고 비실시간 트랜잭션일 경우 FCFS(First Come First Served)[8] 기법을 기반으로 한 큐에 삽입한다. 스케줄러에서는 우선순위 큐에 종료시한 내에 완료가 가능한 트랜잭션들의 집합인 실행 원도우를 생성한 후 실행 원도우

내에 있는 트랜잭션들을 스케줄링한다. 디스패처는 스케줄링된 트랜잭션 중에서 가장 우선순위가 높은 트랜잭션을 선택하는 역할을 한다. 이때 디스패처는 트랜잭션의 접속이 단절되었는지를 검사하여 연결시에만 수행을 시킨다.

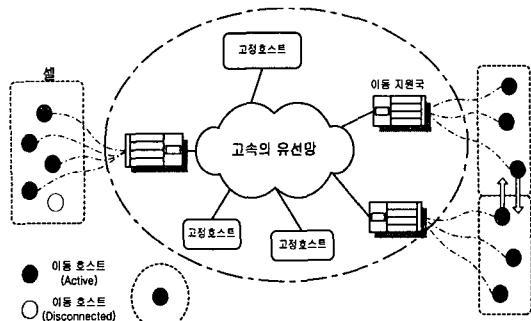
제안된 트랜잭션 관리자는 실행여부 검사를 통해 실시간 이동 트랜잭션을 위한 예측성을 향상시켜 종료시한을 만족하는 실시간 이동 트랜잭션의 수를 최대화할 수 있다. 또한 실시간 데이터의 유효성에 따른 트랜잭션의 자연으로 트랜잭션의 철회를 감소시키며, 이동 호스트와 고정망간의 접속 단절로 인한 전체 트랜잭션의 자연이나 철회를 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서 이동 컴퓨팅 시스템의 특성과 기존의 스케줄링 기법에 대하여 살펴보고, 3장에서는 이동 컴퓨팅 환경에서 실시간 트랜잭션과 실시간 데이터의 제약조건을 준수하면서 사용자가 요구한 트랜잭션의 중요도를 우선으로 하여 실행 원도우를 사용한 트랜잭션 관리자를 설계하며, 4장에서는 마지막으로 본 논문에 대한 결론과 향후 연구 방향에 대하여 논한다.

2. 관련연구

2.1 이동 컴퓨팅 시스템의 특성

이동 컴퓨팅 환경이란 기존의 분산 컴퓨팅 환경이 이동 호스트들을 지원할 수 있도록 구성된 환경을 의미한다[1]. 현재 널리 이용되고 있는 이동 컴퓨팅 환경의 시스템에 대한 구조는 [그림 1]과 같이 이동 호스트(Mobile Host), 이동 지원국(Mobile Support Station), 그리고 고정 호스트(Fixed Host)들로 구성되어 있다[2].



[그림 1] 이동 컴퓨팅 시스템의 구조

이동 호스트는 노트북 컴퓨터 혹은 이동 컴퓨팅 환경에 적합하게 개발된 휴대용 컴퓨터를 사용하며 이동지원국을 통하여 고정 컴퓨터 통신망에 접속할 수 있다. 하나의 이동 지원국이 포괄하는 통신 지역을 무선 통신 셀(cell)이라 하며, 해당 이동지원국은 자신 셀 안에 있는 모든 이동 호스트와의 통신을 관리한다. 이동 지원국은 관리하고 있는 셀 내에 위치한 이동 호스트의 소속 정보, 보안 정보, 위치 정보 등을 자체 관리하고 있으며, 관련 정보를 이용하여 이동 호스트 사용자의 정보 접근을 조정한다. 고정 호스트는 고정 사용자 및 이동 사용자에게 제공할 컴퓨팅 능력 및 정보를 관리한다. 고정 호스트에는 메인 프레임, 워크스테이

션 및 개인용 컴퓨터 등을 용도 및 처리 능력에 따라 설치하여 사용한다.

이동 호스트와 고정호스트의 연결성은 고정망에 안정적으로 연결되어 이동 호스트와 고정망이 원활한 데이터의 전송이 이루어지는 완전 연결 상태(fully connected), 이동 호스트가 이동 지원국의 셀 영역을 벗어나 고정망과 연결이 단절된 상태인 접속 단절 상태(disconnected), 이동 호스트가 셀 영역의 경계지역이나 전파차단 지역 등에 위치하여 고정망과의 이동 호스트간의 접속이 불안전한 상태에 있는 약한 연결 상태(weakly connected), 그리고 사용자가 이동 호스트의 전원의 절약 등을 목적으로 의도적으로 망과의 접속을 분리하는 의도적 접속 단절 상태(spurious disconnected)가 있다[9].

2.2 실시간 트랜잭션 스케줄링 기법

데이터와 시스템의 자원에 대한 충돌을 해결하고, 트랜잭션이 지난 시간적인 제약조건을 만족시키기 위해서 실시간 데이터베이스 시스템에서는 트랜잭션에 우선순위를 부과하는 기법을 사용한다. 이러한 우선순위 기법으로는 가장 먼저 준비된 트랜잭션에 높은 우선순위를 할당하는 기법인 FCFS 기법과 실시간 데이터베이스 시스템의 스케줄링에서 가장 기본적으로 이용되고 있는 기법으로 종료시한에 가장 근접한 트랜잭션에 높은 우선순위를 할당하는 EDF 기법, 그리고 트랜잭션이 자신의 종료시한을 만족시키면서 수행을 얼마나 늦출 수 있는 가를 나타내는 슬랙을 이용하는 LSF 기법[10]을 들 수 있다.

FCFS 기법은 트랜잭션의 종료시한에 관한 정보를 사용하지 않으므로 긴급한 트랜잭션이 도착한 경우에도 먼저 대기하고 있는 트랜잭션이 앞서 처리되므로 긴급한 트랜잭션에 대해서는 대응하기가 어렵다. LSF 기법은 트랜잭션 간에 충돌이 발생할 경우, 높은 우선순위를 가진 트랜잭션이 슬랙시간 내에 처리 가능하다면 낮은 우선순위를 가진 트랜잭션을 철회하지 않고 수행함으로써 자원의 이용을 극대화할 수 있으나, 트랜잭션의 수행시간과 이를 고려한 슬랙 시간을 계산하기가 어려울 뿐만 아니라 EDF 기법에 비해 부하가 크다.

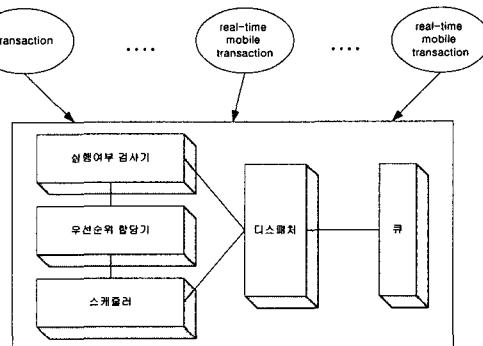
EDF 기법은 종료시한 정보를 사용하므로 긴급한 트랜잭션의 처리에 용이하고, 트랜잭션들이 적당히 적재되는 경우에는 좋은 성능을 유지하지만, 트랜잭션들이 과도하게 적재되는 경우에는 종료시한에 근접하거나 종료시한을 초과하는 트랜잭션에 높은 우선순위가 할당되는 단점이 가지고 있다.

3. 실시간 이동 트랜잭션 관리자의 설계

이 장에서는 이동 컴퓨팅 환경에서 발생되는 실시간 이동 트랜잭션을 처리하기 위한 트랜잭션 관리자를 제안한다. 이동 컴퓨팅 환경을 위한 실시간 데이터베이스 시스템에서는 실제 실시간 트랜잭션과 비 실시간 트랜잭션이 동시에 발생하게 된다. 즉, 시간적인 제약조건을 갖는 실시간 트랜잭션과 평균적인 응답속도를 최소화해야 하는 비 실시간 트랜잭션이 동시에 발생하게 된다. 따라서 실시간 트랜잭션뿐만 아니라 비 실시간 트랜잭션까지 효과적

으로 처리할 수 있어야 한다. 제안된 실시간 이동 트랜잭션 관리자는 [그림 2]와 같다.

디스패처가 트랜잭션 관리자로 실시간 이동 트랜잭션을 가져오면 실행여부 검사기로 보내고, 비 실시간 트랜잭션을 가져오면 큐에 삽입한다. 실행여부 검사기로 보내진 트랜잭션이 실행가능한 트랜잭션이면 우선순위 할당기에서 우선순위를 할당받아 스케줄러를 통해 스케줄링된다. 그 트랜잭션이 스케줄링된 트랜잭션들 중에서 가장 높은 우선순위를 가졌다면 디스패처가 다음 실행될 트랜잭션으로 선택한다. 다음 절들에서 각각의 자세한 사항을 살펴본다.



[그림 2] 실시간 이동 트랜잭션 관리자의 구조

3.1 실행여부 검사기

실행여부 검사기에 도착한 트랜잭션은 먼저 트랜잭션에서 사용하는 실시간 데이터 집합을 사용하여 가장 먼저 데이터의 유효성을 상실하는 값을 계산한다. 계산된 데이터 유효 시한 값을 원래 가지는 트랜잭션의 종료시한과 비교하여 작은 값을 새로운 트랜잭션의 종료시한으로 설정한다.

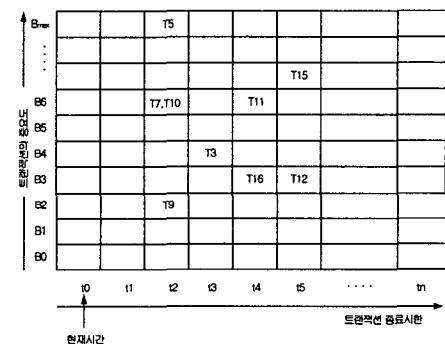
새로 설정된 트랜잭션의 종료시한을 이용하여 트랜잭션이 종료시한 내에 수행 가능한지를 검사한다. 즉, 트랜잭션의 수행 예측시간과 현재 시간을 합하면 트랜잭션의 예상 종료 시간을 산출할 수 있다. 이 값을 종료시한과 비교하여 종료시한의 값이 크면 트랜잭션의 제약조건을 만족시킬 수 있지만 종료시한보다 작은 값이 나오면 트랜잭션의 제약조건을 만족시킬 수 없다. 따라서 전자의 경우는 정상적으로 우선순위 할당기에 있는 우선순위 큐에 삽입되어 트랜잭션이 실행되지만, 후자의 경우는 트랜잭션의 특성에 따라 편 실시간 트랜잭션의 경우에는 트랜잭션을 철회하는 것이 적절하고, 소프트 실시간 트랜잭션의 경우에는 중요도를 낮추어 큐에 삽입한다.

3.2 우선순위 할당기

우선순위 할당기에서는 실시간 데이터베이스 시스템의 우선순위 할당 기법으로 많이 사용되는 EDF 기법을 이용하여 실시간 이동 트랜잭션을 처리한다. 또한, 트랜잭션의 중요도를 기준으로 높은 중요도를 지니는 트랜잭션을 우선적으로 처리하고, 중요도가 낮은 트랜잭션은 대기하는 방식이다. 고정망 시스템의 경우 우선순위 큐를 생성할 때는 하드(hard), 소프트

(soft), 펌(firm)으로 구분되는 실시간 트랜잭션의 유형을 고려했지만, 이동 컴퓨팅 환경에서는 상이한 사용자의 요구가 발생하기 때문에 트랜잭션의 종료시한을 만족시킴으로써 얻을 수 있는 중요도가 더 중요한 의미를 가진다. 그러므로, 본 논문에서 사용하는 우선순위 큐는 [그림 3]과 같이 트랜잭션의 중요도와 종료시한을 나타내는 이차원적인 구조를 가지고 있다. 즉, x축은 트랜잭션의 종료시한을 의미하고, y축은 트랜잭션이 지니는 중요도를 나타내며 최대 값은 Bmax이다.

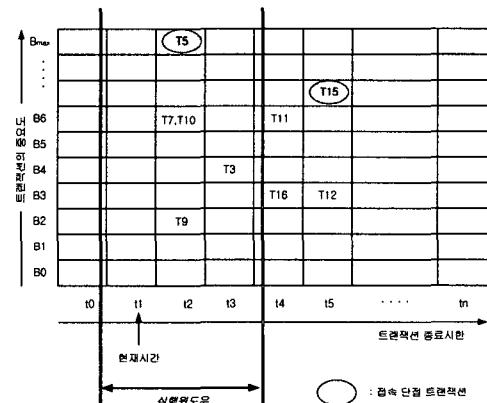
실행여부 검사기에서 보내진 트랜잭션들은 [그림 3]과 같이 우선순위 큐에 배치된다. 우선순위 큐에 트랜잭션을 삽입하는 방법은 트랜잭션의 우선순위와 트랜잭션의 종료 시한을 기준으로 해당 셀에 할당하는 것이다. 예를 들어, 트랜잭션 T3이 B4의 중요도를 가지고 있고, 설정된 종료 시한은 t3 값이라면 셀(t3, B4)를 할당 받는다.



[그림 3] 우선순위 큐

3.3 스케줄러

스케줄러가 스케줄링 할 트랜잭션을 선택하기 위해 본 논문에서는 실행 원도우 기법을 사용한다. 실행 원도우는 현재 시간, 트랜잭션의 수행 정보, 중요도, 그리고 종료시한을 고려했을 때 종료시한 내에 수행 가능한 트랜잭션의 집합을 의미하고, 스케줄러는 실행 원도우 내에 있는 트랜잭션들을 대상으로 스케줄링을 한다.



[그림 4] 운선순위 큐에서의 실행 외도운

[그림 4]는 현재 시간이 t1일 때, 실행 윈도우가 설정된 예를 보여 주고 있다. 실행 윈도우 안에 트랜잭션 T3, T5, T7, T9, T10 가 속해있고, 스케줄러는 중요도가 가장 높은 트랜잭션 T5에 가장 높은 우선순위를 할당한다. 즉, 실행 윈도우 내에서 가장 가중치가 높은 트랜잭션을 우선적으로 처리는 기법으로 [그림 4]의 예에서는 T5 → T7, T10 → T3 → T9 순서로 우선순위를 할당한다. 이 기법을 사용하면 전체적인 트랜잭션의 중요도의 합을 최대화 할 수 있다.

3.4 디스패처

디스패처는 두 가지 역할을 담당한다. 첫째, 트랜잭션의 형태 즉, 실시간 트랜잭션과 비 실시간 트랜잭션에 따라 실시간 트랜잭션은 실행여부 검사기를 거쳐 우선순위 큐에 할당하고, 비 실시간 트랜잭션은 큐에 할당하는 역할을 담당한다. 둘째, 수행 중인 트랜잭션이 완료되었을 때, 다음에 수행할 트랜잭션을 스케줄러에서 선택하는 역할을 한다. 두 번째 역할을 수행할 때는 이동 호스트의 특성에 따른 빈번한 접속 단절을 고려해야 한다.

접속 단절이 발생한 트랜잭션의 처리는 이동 호스트가 연결될 때까지 기다리므로 트랜잭션의 지역 현상이 발생하여 동시성을 저하시킬 수 있다. 따라서 실행 윈도우 기반의 스케줄링 기법에서는 접속 단절이 발생한 트랜잭션의 경우 해당 스케줄링 대상에 제외하고 수행 가능성이 높은 트랜잭션을 우선적으로 실행한다.

[그림 4]에서 T5 와 T15는 이동 호스트가 접속 단절인 상태이다. 따라서 스케줄러가 스케줄링한 T5 → T7, T10 → T3 → T9에서 T5 가 높은 중요도를 갖는 트랜잭션이지만 접속 단절이 발생하였으므로 디스패처는 T5 가 아닌 T7이나 T10을 선택하여 실행한다.

4. 결 론

본 논문에서는 호스트가 자유롭게 이동하는 이동 컴퓨팅 환경에서 시간 제약 조건을 지니는 실시간 이동 트랜잭션을 지원하기 위한 트랜잭션 관리자에 관한 연구를 수행하였다. 이동 컴퓨팅 환경에서 이동 호스트의 이동성에 의한 접속 단절과 핸드오버 등에 의해 트랜잭션의 실행에 대한 예측성이 저하됨으로 실시간 특성을 지원하는 것이 매우 어렵다. 또한, 기존의 종료시한을 갖는 실시간 트랜잭션 처리를 위한 스케줄링 기법, 동시성 제어 기법 등이고 경망을 기반으로 개발된 것으로 이를 무선통신을 사용하는 이동 컴퓨팅 환경에 적용하는 것은 부적합하다. 따라서 본 논문에서는 이동 컴퓨팅 환경에서 발생하는 실시간 이동 트랜잭션을 처리하기 위한 실시간 이동 트랜잭션 관리자를 설계하였다.

제안된 실시간 이동 트랜잭션 관리자는 트랜잭션의 종료시한과 실시간 데이터의 유효성을 고려하여 실행여부를 판단하여 예측성을 향상시켰으며, 트랜잭션의 중요도와 종료시한을 기반으로 한 우선순위 큐에 트랜잭션을 배치시켜 우선순위를 할당했다. 또한, 스케줄링을 위해서 우선순위 큐에 최적의 실행 윈도우를 생성하고 이를 이용하여

실행될 트랜잭션을 선택하였다. 따라서 트랜잭션 관리자는 기존에 고려하지 않았던 실시간 데이터의 유효성 제약조건 위반에 따른 트랜잭션의 제시작을 방지할 수 있었으며, 종료시한을 만족하는 트랜잭션의 중요도의 합을 최대로 한다.

향후 연구 방향으로는 실시간 이동 트랜잭션의 회복 기법에 대한 연구가 진행되어야 한다. 실시간 이동 트랜잭션의 회복은 검사점 기법을 기반으로 데이터에 따라 서로 다른 기법의 회복연산을 적용해야 한다. 특히, 실시간 데이터의 경우에는 유효시간이 경과된 경우에 트랜잭션의 회복은 기존의 작업을 철회하면 된다. 따라서 향후 본 논문에서 제안한 스케줄링 기법을 이용한 트랜잭션의 회복에 대한 연구가 필요하다.

[참고 문헌]

- [1] Tomasz Imielinski and B. R. Badrinath, "Wireless mobile computing: Solutions and challenges in data management," Technical Report DCS-TR-296, Department of Computer Science, Rutgers University, New Brunswick, NJ 08903, 1993.
- [2] M.H. Dunham and A. Helal, "Mobile computing and databases: Anything new?", ACM SIGMOD Record, 24(4): pp.5-9, December 1995.
- [3] O. Ulusoy, "Real-Time data management for mobile computing," IADT'98, pp.223-240, 1998.
- [4] E. Pitoura and B. Bhargava, "Maintaining consistency of data in mobile distributed environments," Proceedings of 15th International Conference on Distributed Computing Systems, 1995.
- [5] 김경배, 배해영, "분류된 클래스 큐를 이용한 실시간 데이터베이스 시스템의 트랜잭션 관리기," 한국정보처리학회 논문지, Vol.5, No.11, 1998.
- [6] G.D. Walborn and P.K. Chrysanthis, "Supporting semantics-based transaction processing in mobile database applications," In Proceedings of the 14th IEEE Symposium on Reliable Distributed Systems, September 1995.
- [7] J. R. Haritsa, M. Livny and M. J. Carey, "Earliest Deadline Scheduling for Real-Time Database Systems," Proceedings 12th Real-Time System Symposium, IEEE, pp.232-242, 1991.
- [8] R. Abbott and H. Garcia-Molina, "Scheduling I/O Requests with Deadlines : a Performance Evaluation," Proceedings of 11th Real-Time Systems Symposium, IEEE, pp.113-124, 1990.
- [9] P.K. Chrysanthis, "Transaction processing in mobile computing environment," In IEEE Workshop on Advances in Parallel and Distributed Systems, pp.77-82, October 1993.
- [10] R. Abbott and H. Garcia-Molina, "Scheduling Real-time Transactions," SIGMOD RECORD, ACM, Vol.17, No.1, March 1988.