

# 객체 관계형 실시간 DBMS Tachyon

박유미<sup>0</sup>, 배명남, 김원영, 한미경, 유한양, 최 완, 김해근, 조주현\*  
한국전자통신연구원, ㈜코스모\*  
{parkym, mnbae, wykim, mkhan, hyyou, wchoi}@etri.re.kr, hgkim@nice.etri.re.kr,  
jhcho@cosmo.re.kr\*

## Tachyon, the Object-Relational Real-time DBMS

Yoo-mi Park<sup>0</sup>, Myung-nam Bae, Won-young Kim, Mi-kyoung Han,  
Han-yang You, Wan Choi, Hae-geun Kim, and Ju-hyun Cho  
ETRI

### 요 약

본 논문은 객체 관계형 실시간 DBMS 인 Tachyon의 특성 및 구조에 관하여 기술한다. Tachyon은 전전자 교환기용 관계형 DBMS인 DREAM-S의 개발 경험을 바탕으로 개발된 객체 관계형 DBMS이고, 고성능 데이터 처리를 위한 주기억 장치 상주 DBMS이며, 우선순위 기반의 스케줄링과 연성 데드라인(soft-deadline) 트랜잭션을 처리하는 실시간 DBMS이다. Tachyon은 객체지향 질의를 통한 융통성 있는 데이터 조작성이 가능하며, 다중 사용자 환경에서 항상 데이터의 일관성을 유지하고, 고신뢰 시스템을 위한 이중화 구조의 지원 및 데이터 백업/복구 등의 종합적인 데이터베이스 관리를 제공한다. 현재 Tachyon은 DREAM-S와 더불어 초고속정보통신망에서의 백본 교환기 시스템의 DBMS로 활용될 예정이다.

## 1. 서론

최근 초고속 정보 통신망을 기반으로 한 인터넷 서비스가 활성화됨에 따라 높아지는 데이터 전송속도에 비례하여 서비스 제공 시스템의 데이터 처리량이 많아지고 있다. 그러나, 현재 이러한 데이터를 관리하고 있는 대부분의 디스크 기반 DBMS는 DBMS 자체의 기능은 방대하나 규모가 크고, 데이터 처리 성능이 떨어져 인터넷 상에서 폭주하는 데이터를 처리하기에 속도 및 성능 측면에서 부족한 점들이 지적되고 있다. 이러한 현실에서 RAM가격의 하락에 힘입어 고성능 주기억 장치 상주형 DBMS가 전 세계적으로 시장 점유율을 높여가고 있는 추세이다[1].

Tachyon은 10여년간의 국산 전전자 교환기용 주기억장치 상주 DBMS인 DREAM-S[2]의 개발 경험을 바탕으로 관계형 DBMS의 장점을 살리고 단점을 보완하여 설계 및 구현된 객체 관계형 DBMS로서, 우선순위 기반의 스케줄링과 연성 데드라인(soft-deadline) 트랜잭션을 처리하는 실시간 DBMS이다. Tachyon은 고성능 데이터 접근 및 처리를 위한 주기억 장치 상주 DBMS이며, 객체식별자(OID), 클래스 및 질의의 상속, 그리고 사용자 정의 데이터 타입 등을 지원하는 객체 관계형 DBMS로서 객체 식별자를 이용한 효율적인 데이터 검색을 수행하고, 다양한 사용자 인터페이스를 지원한다. 또한 DBMS의 이중화를 통한 고장감내형 고신뢰 구조를 가지는 특징이 있다.

본 논문에서는 이와 같은 Tachyon의 특성 및 기능과 구조에 관하여 기술하며 본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구에 관하여 기술하고, 제 3장에서는 Tachyon이 지원하는 객체 관계형 데이터 모델을 중심으로 Tachyon의 특성을 기술하고, 제 4장에서는 Tachyon 서버의 구조 및 기능과 사용자 인터페이스에 대하여 기술하고, 마지막으로 제 5장에서 결론을 맺는다.

## 2. 관련 연구

내장형 DBMS 및 범용 디스크 기반 DBMS 시장의 front-tier로서 전 세계적으로 주기억장치 상주형 DBMS의 개발이 활기를 띠고 있다. 이에 대표적인 DBMS 들로는 미국의 TimesTen[3]과 RAIMA[4], 영국의 Polyhedra[5] 등을 꼽을 수 있다. 본 장에서는 상기 DBMS들과 국산 전전자 교환기용 DBMS인 DREAM-S[2]의 특성에 관하여 비교 기술한다.

TimesTen[3]은 미국 TimesTen사의 관계형 주기억장치 상주 DBMS로 주기억장치 상주로의 데이터 접근을 고려한 질의 최적화 알고리즘, 버퍼 풀 관리, 인덱싱 구조 등으로 DBMS 성능을 높였다. 현재 디스크 기반의 관계형 DBMS의 데이터베이스 캐싱(caching) 기능을 수행하는 front-tier 제품이 디스크 기반 관계형 DBMS 상에서 10배 이상의 데이터 처리 성능을 보이고 있다고 발표하고 있다. 미국의 RAIMA[4]는 상용 실시간OS를 탑재한 내장형 시스템 시장을 겨냥한 관계형 모델과 네트워크 모델을 혼합한 모델을 지원하는 주기억장치 상주 DBMS로서 SQL보다는 API를 이용한 데이터 접근 방법에 주력하고 있다.

영국에서 개발된 Polyhedra[5]는 실시간 응용을 위한 내장형 시스템을 위하여 개발된 객체 관계형 주기억장치 상주 DBMS로서 event-driven 방식의 사용자 정의 메소드 호출과 active query 를 지원한다. 또한 active 와 standby로 구분되는 이중의 데이터베이스 구조를 가지고 있다. DREAM-S[2]는 국산 전자 교환기용 데이터를 관리하기 위하여 개발된 관계형 주기억장치 상주 DBMS로서 SROS라는 교환기용 실시간OS상에서 동작하며 다양하고 직접적인 메모리 접근 방법과 미리 컴파일된 응용 프로그램을 이용해 데이터 접근 성능을 높였다.

### 3. Tachyon 특성

#### 가. 특성

Tachyon의 주요 특징으로는, 1) 데이터베이스를 주기억장치에 상주시키고, 주기억장치 DBMS에 가장 적합한 인덱싱 방법으로 알려진 T-tree 인덱싱을 지원하고 있어, 빠른 응답시간을 제공할 수 있다. 또한, 2) DBMS 엔진 수준에서 객체관계형 모델을 지원하고 있다[6-8] 주요 개념으로는 객체, 클래스, 상속, 사용자 정의 타입을 지원하며, 객체들간의 다양한 관계성을 명시하기 위해, OID\_REF, OID\_SET, INVERSE 구조를 함께 제공하고 있다. 특히, 이들을 통해 관계형 모델에서의 조인(join) 연산 시 발생하는 부가적인 메모리의 할당 및 응답시간 지연을 최소화 할 수 있다. 다음으로 3) 표준 DBMS 접근 체계인 ODBC 3.0 규격과 JDBC를 지원하여, C/C++, java, VB, CHILL/OOCHILL등의 다양한 언어를 지원한다. 그리고, 4) solaris, SROS, Windows NT와 같은 여러 운영체제 플랫폼을 지원하고 있으며, 고도의 안정성을 요구하는 응용을 위해서는 운영체제와 별도로 자체의 이중화 방안을 제공하고 있다.

#### 나. 데이터 모델

Tachyon의 데이터 모델은 관계형 데이터 모델에 사용자들이 친숙한 객체지향 프로그래밍 기법을 확장하여 정의하였다.

Tachyon에서, 클래스는 단위 의미를 가지며 동일한 형태의 객체(object, or entity)들의 집합을 의미한다. 객체는 시스템에서 유일하며, 객체의 물리적인 위치를 갖는 식별자(OID, Object Identifier)에 의해 구분된다.

클래스간의 수평적인 관계성은 객체 내에 관련된 객체식별자를 속성으로 포함함으로써 성립된다. 예를 들어, 클래스 C<sub>1</sub>, C<sub>2</sub>간에 관계성 R<sub>c1-c2</sub>이 존재한다면, 관계성이 있는 두 객체 o<sub>c1</sub>∈C<sub>1</sub>, o<sub>c2</sub>∈C<sub>2</sub>는 각각 o<sub>c2</sub>, o<sub>c1</sub>의 식별자를 속성으로 갖게 된다. Tachyon에서는 스키마에 이러한 관계성과 관계성의 다중성(multiplicity)을 명시하기 위해, OID\_REF, OID\_SET 및 inverse를 정의하였으며, 이를 사용하여 정의한 실 예는 다음과 같다.

예1) 클래스와 관계성의 정의

```

Create class C1 {
  Rc1,c2 OID_REF C2,
  Attrc1
}
create class C2 {
  Rc2,c1 OID_REF inverse C1.Rc1,c2,
  Attrc2
}
    
```

관계성을 명시할 때, 객체식별자를 사용하는 이유는 C<sub>1</sub>과 C<sub>2</sub>간의 조인이 요구될 경우 예2)와 같이 R<sub>c1,c2</sub>(혹은 C<sub>2</sub>의 R<sub>c2,c1</sub>)로 직접 접근할 수 있기 때문이다. 이 경우에, 임시 테이블을 생성하기 위한 부가적인 저장장소나 시간의 사용 없이, 조인 결과를 직접 얻을 수 있다.

예2) 관계성을 사용한 조인

```

select Attrc1, Rc1,c2-> Attrc2
from C1 ;
=
select Rc2,c1-> Attrc1, Attrc2
from C2 ;
    
```

클래스간의 수직적인 관계성(일반화)은 이미 정의한 구조 및 관계성의 재사용 관점을 지원하며, 일반화를 통해 상속된 구조에 대한 질의 상속도 함께 지원한다. 예를 들어, C<sub>1</sub>에서 상속 받은 클래스 C<sub>3</sub>를 고려해 보자.

예3) 클래스 상속의 정의

```

create class C3 under C1 {
  Rc1,c2 OID_REF C2, # C1으로 부터 상속
  Attrc1, # C1으로 부터 상속
  Attrc3
}
    
```

클래스 C<sub>3</sub>는 C<sub>1</sub>의 구조를 상속받으므로(C<sub>1</sub>과 C<sub>2</sub>간의 관계성까지도 상속), o<sub>c3</sub>∈C<sub>3</sub>와 o' <sub>c2</sub>∈C<sub>2</sub>간에 R<sub>c1,c2</sub>로 연관성을 정의할 수 있으며, 역시 C<sub>3</sub>에 대해 예2)와 같은 질의도 가능하다. 또한, C<sub>3</sub>는 C<sub>1</sub>에서 상속받은 구조를 갖고 있기 때문에, C<sub>1</sub>에 대한 질의는 상속관계에 있는 C<sub>3</sub>에게 그대로 상속되어 실행되고, 그 결과는 C<sub>1</sub>에 대한 질의 결과에 union되어 사용자에게 제공된다. 상속을 정의한 후, 예2)의 질의가 적용될 경우의 아래와 같이 확장되어 실행된다.

예4) 클래스 계층 구조에 따른 질의 상속

```

select . Attrc1, Rc1,c2-> Attrc2
from C1 ;
=
select Attrc1, Rc1,c2-> Attrc2
from C1 ;
UNION
select Attrc1, Rc1,c2-> Attrc2
from C3 ;
    
```

이를 통해, 상속받은 모든 구조에 대해 단일의 질의로 연산이 가능하다. 이와 같은 클래스 구조의 상속과 객체식별자를 사용한 관계성 정의를 통해, 기존의 관계형 모델의 고비용 연산인 조인을 대체할 수 있으며, 응용 데이터 모델과의 유사성으로 보다 자연스러운 데이터 모델링이 가능하다는 장점이 있다.

### 4. Tachyon 구조

Tachyon은 그림 1과 같이 클라이언트-서버 구조를 기반으로 한다. 서버는 다중 쓰레드로 구성되며, 클라이언트들의 요청을 하나 이상의 쓰레드들이 동시에 처리한다. Tachyon 서버는 실시간 응용을 위해 우선순위 큐를 지원하며 트랜잭션 스케줄링에 있어 우선순위 역행 현상을 보완하기 위해 트랜잭션에 대한 우선순위 상속(priority inheritance)[9]을 수행할 뿐만 아니라, 교착상태 처리 시 우선순위가 낮은 트랜잭션을 희생자로 선정한다.

Tachyon 서버는 클래스 계층구조에 따른 질의를 지원하며, OID를 이용한 커서 향대로 조인을 대체함으로써 질의 처리 성능을 높일 수 있도록 설계되었다. 또한, 사용자 데이터나 시스템 카탈로그 관리를 위해 Tachyon에서 제공하는 객체관계형 구조를 이용한다. 그리고 사용자 접근 처리를 위해 Tachyon 서버는 메모리 상주형 DBMS에서 최적의 성능을 보이는 T-tree 인덱스를 구성하여 관리한다.

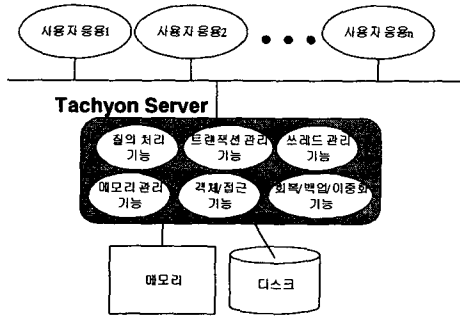


그림 1. Tachyon의 구조

Tachyon은 32bit 운영체제의 경우 1MB부터 4GB 크기의 데이터베이스를 지원할 수 있다. 데이터베이스의 회복을 위해서는 논리적 로깅과 물리적 로깅을 병행하여 신뢰성 있는 로그 관리를 지원하고, 백업을 위해서는 성능을 고려한 퍼지 체크포인팅 방식[10]을 사용한다. 또한, 높은 신뢰성을 요구하는 교환기와 같은 구조에서 사용될 수 있도록 고장 감내를 위한 이중화 구조를 제공한다.

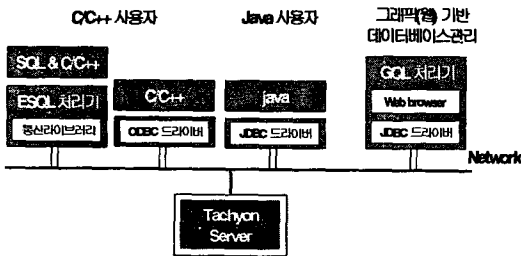


그림 2. Tachyon의 사용자 인터페이스

그림 2는 Tachyon 개발 환경으로, Tachyon SQL을 호스트 프로그래밍 언어에 삽입하여 사용하는 ES/SQL을 지원하며, 웹 응용을 지원하기 위하여 ODBC3.0, JDBC 2.0를 지원한다. ES/SQL이 사용 가능한 언어는 현재 C, C++, CHILL, OO/HILL이며, JDBC를 통하여 JAVA에서도 프로그램도 Tachyon의 데이터베이스로의 접근도 지원한다. 또한 PC 사용자를 위하여 그래픽 질의 처리기(GQL:Graphic Query Language)도 제공된다.

### 5. 결론

본 논문에서는 국내에서 처음으로 개발된 객체 관계형 실시간 DBMS인 Tachyon의 구조 및 특성에 대해 서술하였다. 서술한 바와 같이 Tachyon은 메모리 상주 데이터베이스로 고성능을 제공하며, 우선순위 기반과 soft-deadline 트랜잭션 처리 방식을 제공하는 실시간 DBMS이다.

Tachyon은 셋째, 실제계를 반영하기에 용이한 객체 관계형 모델을 지원하여, 응용 프로그램의 설계, 개발을 용이하게 할 뿐만 아니라, 재사용과 상속 및 사용자 정의 타입 지원 등의 특성을 제공할 수 있도록 설계하였다. 둘째, 객체지향형의 특성을 이용한 포인터 기반의 질의를 처리할 수 있으므로, 기존의 RDBMS에서의 조인 연산을 대체하여 우수한 성능을 제공할 수 있는 특성을 제공한다.

셋째, OID\_REF, OID\_SET, INVERSE 등의 다양한 관계성을 표현할 수 있다. 넷째, Tachyon은 C/C++, VB 및 CHILL 언어 사용자를 위한 ES/SQL과 Web 응용 프로그램 개발자를 위한 ODBC/JDBC 인터페이스를 제공하며, 그래픽 기반의 대화형 질의처리 도구 등의 다양한 사용자 인터페이스를 제공한다. 다섯째, Tachyon은 Solaris를 기반으로 Linux, Windows NT 및 실시간 OS인 SROS, VxWorks 등의 다중 플랫폼을 지원할 수 있도록 개발되었다. 또한, Tachyon은 이중화 기능을 지원함으로써 고장 감내의 특성을 제공하여 신뢰성을 제공하므로 다양한 응용에 사용될 수 있다.

그러나, Tachyon은 다양한 관계성 표현 방식과 객체지향형의 특성을 제공하므로, 복잡한 데이터베이스 모델링을 용이하게 할 수 있는 자동화 도구가 절실히 요구된다. 따라서, UML 등을 이용한 데이터베이스 모델링 방식의 자동화 도구를 제공할 수 있도록 이에 대한 연구를 우선적으로 진행할 예정이다. 또한, 기존의 관계형 DBMS 및 객체지향형 DBMS에 대한 벤치마크 자료를 분석하여 신뢰성 있는 객체 관계형 DBMS의 성능 평가도구를 개발하여 Tachyon의 벤치마크 결과를 도출하고 Tachyon의 성능 향상 및 안정화를 위한 연구가 계속 진행되어야 할 것으로 사료된다.

### 6. 참고 문헌

- [1]. H.Garcia-Molina and K.Salem, "Main Memory Database Systems: An Overview," IEEE TKDE, Vol. 4, No. 6, pp. 509-516, Dec. 1992.
- [2]. Y. Yoon, et.al., "Scalable Distributed Real-time Database Management for Switching System," ISS 97, pp539-545, Sep. 1997.
- [3]. TimesTen Performance Software, *In-Memory Data Management Technical White Paper*, White paper, TimesTen Performance Software.
- [4]. Centura Software Co., *Raima Database Manager++ In Real-time and Embedded Systems*, White Paper, Centura Software Co..
- [5]. Polyhedra Plc., *Bloor Research Overview of Polyhedra*, White Paper, Polyhedra Plc.
- [6]. M. Stonebreaker and P. Brown, "Object-Relational DBMSs : Tracking the Next Great Wave," San Francisco: Morgan Kaufmann Publishers, 1999.
- [7]. C. Maro Saracco, "Universal Database Management: A Guide to Object/Relational," San Francisco: Morgan Kaufmann Publishers, 1998.
- [8]. C. T. Yu; W. Meng, "Principles of Database Query Processing for Advanced Applications," San Francisco: Morgan Kaufmann Publishers, 1998.
- [9]. L. Sha, R. Rajkumar, and J.P. Lehoczky, "Priority Inheritance Protocols: An Approach to Real-time Synchronization," IEEE TOC, Vol.39, No.9, pp.1175 - 1185, 1990.
- [10]. X.Li and M.H.Eich, "Post-crash Log Processing for Fuzzy Checkpointing Main Memory Databases," Proc. ICDE, pp. 117-124, 1993.