특집

Flash Memory 기반 데이터베이스 연구동향

나갑주 · 이상원 (성균관대학교)

I. 서 론

데이터베이스의 사전적인 의미는 다수의 사용자에 의해 공유되어 사용되는 통합적인 데이터의 집합을 의미하며 데이터베이스 관리 시스템은 다수의 사용자들이 데이터베이스 내에 효과적으로 접근 할 수 있도록 해주는 소프트웨어를 의미한다. 따라서 데이터베이스 관리 시스템의 일차적인 목적은 얼마나 빨리 원하는 데이터를 저장 장치에 입력시킬 수 있는지와 저장된 데이터를 얼마나 빠르게 찾아낼 수 있느냐가 된다. 결국 데이터가 저장되는 저장장치의 특성을 얼마나 잘 이해하고 설계되는지에 따라 데이터베이스 관리 시스템의 성능이 결정된다고 볼 수 있다.

지난 수십 년간 저장 장치시장은 하드디스크에 의해 지배되어 왔다. 하드디스크는 기계적인 장치를 기반으로 입출력 동작을 수행하고 있기 때문에 순차적인 접근이 매우 뛰어난 장점을 가지고 있다. 반면 임의적인 접근(random access)에는 치명적인 단점을 보이고 있다. 따라서 데이터베이스 관리 시스템의 저장 관리자는 점점 더 데이터에 대한 순차적인 접근을 효과적으로 처리하는 방향으로 설계되어 왔다.

그러나 최근 기계적인 장치에 의존하지 않고 임의 접근이 뛰어난 플래시메모리의 등장으로 인해 순차접근 위주의 기존 설계 방식에 대한 재고찰을 요구되고 있다. 이에 본 고에서는 플래시메모리를 기반으로 하는 데이터베이스 관리 시스템 연구동향에 대해 살펴보겠다. II장 에서는 플래시메모리와 플래시메모리 기반의 저장장치인 Solid State Disk(이하 SSD)의 특성에 대해 간략하게 설명하고 III장에서 플래시메모리의 특성을 직접적으로 활용하는 데이터베이스 관리 시스템의 새로운 저장 관리 기법인 In-Page Logging(IPL)기법에 대해 설명한다. 또한 IV장에서는 기존의 데이터베이스 시스템에서 SSD를 효과적으로 사용하기 위한 연구들에 대해 소개한다. 마지막으로 V장에서는 최근 시도되고 있는 다양한 데이터베이스 분야의 플래시메모리 응용 기법에 대해 소개를 통해 결론을 맺는다.

II. 플래시메모리와 Solid State Disk

플래시메모리는 비휘발성 저장장치인 EEPROM의 일종으로 크게 내부적인 소자에 따라 NOR와

NAND 형태로 나뉜다. NOR 플래시는 비교적 적은 양의 데이터를 저장할 수 있으며 주로 코드 수행을 위한 용도로 활용되며 NAND 플래시의 경우 데이터 저장 장치로써 주로 활용되고 있다.

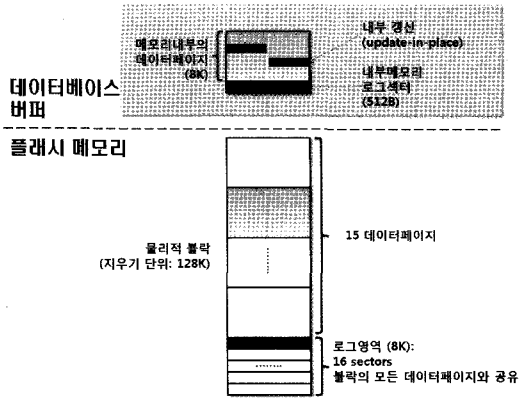
플래시메모리의 가장 큰 특징은 전기적인 신호만을 사용하고 있기 때문에 디스크 기반의 저장장치에 비해 월등하게 뛰어난 접근 속도와 저전력 및 강한 내구성 등의 장점을 가지고 있다. 또한 디스크와 달리 읽기/쓰기/지우기의 세 가지 연산으로 동작되며 덮어쓰기 연산이 불가능하기 때문에 기존에 데이터가 저장된 영역에 쓰기 연산을 수행하기 위해서는 지우기 연산을 수행한 후에 쓰기 연산을 수행하여야 하는 단점이 있다. 또한 읽기/쓰기의 연산 단위와 지우기 연산의 단위가 다르기 때문에 덮어쓰기 연산에 대한 별도의 고려가 요구된다. 그 밖에 지우기 연산의 경우 단위 영역에 대한 지우기 연산 횟수가 제한되는 단점도 가지고 있다.

따라서 NAND 플래시 메모리를 직접 저장 장치로 쓰기 위해서는 기존의 소프트웨어의 재설계가 요구되며 이를 해결하기 위해 Flash Translation Layer (이하 FTL) 이라 불리는 일종의 시스템 소프트웨어를 활용하여 플래시메모리의 내부 연산을 숨기고 기존의 하드 디스크와 동일한 인터페이스를 활용하여 플래시 메모리 기반의 저장장치를 설계한다. 대표적인 플래시메모리 기반의 저장장치로는 SSD 등이 있으며, 최근 다양한 영역에서 하드 디스크를 대체하고 있다.

SSD의 연산은 읽기/쓰기 기반의 하드 디스크와 동일하기 때문에 기존의 소프트웨어의 재설계 없이 사용이 가능하며 내부에 탑재된 FTL에 따라 성능 및 동작 특성이 다르게 나타난다. 그러나 근본적인 NAND 플래시메모리의 특성으로 인해 몇 가지 공통적인 특성이 나타난다. 먼저 SSD에서는 하드 디스크와 달리 임의 읽기 연산이 대단히 빠른 특성을 가지고 있다. 반면, 덮어쓰기가 지원되지 않는 플래시메모리 칩의 특성으로 인해서 임의쓰기연산이 비효율적일 수 있다. 하지만, 임의쓰기 연산의 경우도 FTL 및 SSD 기술의 발전으로 인해서 상당히 개선된 성능을 제공하는 제품들이 시장에 등장하고 있다.

III. In-Page Logging 방식

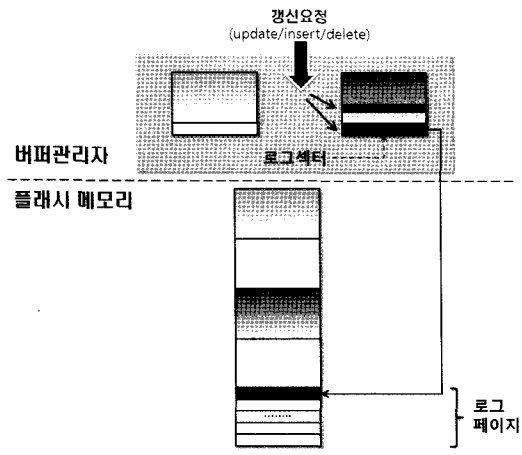
디스크 기반으로 설계된 일반적인 어플리케이션들의 경우 디스크의 기본 입출력 단위인 블록 혹은 섹터단위로 데이터에 대한 읽기/쓰기 연산을 수행한다. 반면, 데이터베이스의 기본 연산 단위는 레코드 혹은 튜플 단위로 이루어지며 서로 논리적인 관계를 가지고 있다. 따라서 데이터베이스 관리 시스템의 경우 물리적인 섹터 혹은 블록에 기록될 데이터의 변화를 적은 양의 논리적인 관계로 표현이 가능하며 이를 로그의 형태로 관리한다. 일반적으로 데이터베이스 관리 시스템에서는 빠른 응답속도 및 다수의 사용자에 대한 동시성 그리고 고장 관리를 위한 용도로 로그를 사용하여 데이터의 입/출력 성능을 향상시키고 있다. In-Page Logging (이하 IPL) 은 데이터베이스 관리 시스템의 로그 관리 기법을 플래시메모리의 특성을 효과적으로 활용할 수 있게 설계된 플래시메모리 기반의 데이터 저장 관리 기법이다. <그림 1>은 IPL의 기본 구조를 나타내고 있다. 그림과 같이 플래시메모리의 지우기 연산 단위인 물리적 블록은 데이터를 저장하는 데이터 페이지 영역과 데이터 페이지에 대한 로그(변화 정보)를 기록하는 로그 영역으로 구분된다. 또한, 데이터베이스 버퍼의 기본 단위인 페이



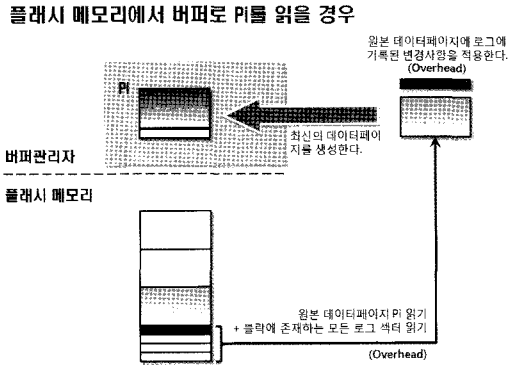
〈그림 1〉 IPL의 기본 구조

지 역시 데이터를 저장하는 데이터페이지와 페이지 별로 할당되는 내부 로그 섹터(In-Page Log Sector)로 구분된다.

만약 <그림 2>와 같이 버퍼에 위치한 데이터페이지에 대한 갱신 연산이 발생한다면 버퍼에 위치한 페이지는 내부 갱신(in-place update)이 수행되며 동시에 갱신에 대한 로그 정보를 내부 로그 섹터에 기록한다. 이후 로그 섹터가 꽉 차거나 데이터베이스 버퍼의 메모리 관리 정책에 의해 저장 장치로 기록이 되어야 한다면, IPL 저장 관리자는 페이지가 이전에 기록된 적이 없을 경우 블록의 데이터페이지에 쓰기 연산을 수



〈그림 2〉 IPL의 쓰기 연산



〈그림 3〉 IPL의 읽기 연산

행하고 이미 데이터페이지가 기록되었다면 내부 갱신이 불가능한 플래시메모리의 제약을 피 적이위해 버퍼에 위치한 페이지 대신 로그 섹터만을 물리적 블록의 로그 영역에 기록하게 된다. 따라서 그림과 같이 8KB의 페이지, 쓱기 연산 대신 512 Byte, 쓱기 연산이 수행됨으로써 쓰기 연산을 최소화 시킬 수 있게 된다. 한편펜에 위치한 페에 기록된 데이터페이지에 대한 읽기 IPL <그림 3>과 같이 물리적인 블록에 기록된 데이터페이지에 대한 읽기 연산과 더불어 해당 페이지와 연관된 로그 섹터들에 대한 읽기 연산을 수행한다. 읽어 들인 로그 섹터에 기록된 로그들을 페이지에 적용함으로써 버퍼에는 항상 최신 페이지 제공할 수 있게 된다.

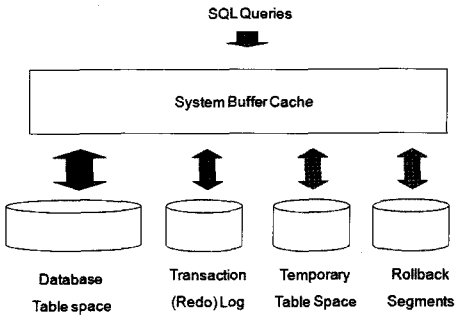
결국 IPL Scheme은 데이터베이스의 로그의 특징을 활용하여 플래시메모리의 내부 갱신연산을 회피하고 플래시메모리의 빠른 읽기 연산을 활용하여 데이터베이스 관리 시스템의 입/출력을 관리할 수 있게 된다.

IV. SSD를 활용한 데이터베이스

IPL 기법은 플래시메모리를 기반으로 데이터

베이스 관리 시스템을 설계하는 일종의 플래시 저장 기법이다. 따라서 IPL을 실제 적용하기 위해서는 기존의 디스크 기반의 데이터베이스 시스템에 대해 전체적인 재설계가 불가피하다. 또한 기존의 디스크 기반의 데이터베이스와 같은 안정성을 보장 받기 위해서는 지속적인 연구가 요구된다. 이와 달리 플래시메모리를 기반으로 한 SSD의 등장으로 인해 최근 활발하게 진행되고 있는 데이터베이스 분야의 연구는 기존의 데이터베이스 시스템의 하드 디스크를 SSD로 대체할 경우 발생되는 가격대 성능 비용에 대한 분석을 통해 SSD의 적용에 대한 효율성에 관련된 연구가 주를 이루고 있다. 즉, SSD의 뛰어난 성능에도 불구하고 여전히 높은 가격으로 인해 엔터프라이즈 데이터베이스 시장에서 SSD는 잠재력을 지닌 저장 장치로써만 인식되어 왔으며 이로 인해 데이터베이스의 모든 하드 디스크를 교체하는 대신 소수 디스크를 SSD로 교체함으로써 성능 향상과 가격 경쟁력이 뛰어난 시스템을 구축하기 위한 연구들이 진행되고 있다. 본 절에서는 데이터베이스 분야에 SSD를 적용함으로써 획기적인 성능 향상을 실증적으로 제시한 최근 연구 [2,3] 내용을 중심으로 설명하겠다.

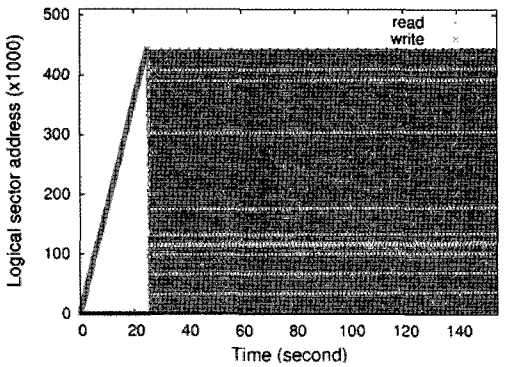
<그림 4>는 데이터베이스 관리 시스템에서 사용되는 하드 디스크의 구조를 개략적으로 나타내고 있다. 그림과 같이 데이터베이스 관리 시스템은 성능과 안정성 및 복구의 효율성을 위해 다수의 물리적 저장장치를 활용하고 있다. 이 중 "데이터베이스 테이블스페이스" 는 실제 데이터가 물리적으로 저장되는 영역으로 가장 큰 용량을 차지한다. 그림과 같이 사용자의 요청이 따른 SQL 질의가 들어올 경우 데이터베이스 버퍼에 원하는 데이터를 읽기 위해 데이터베이스 테이블스페이스 영역에 대한 입출력 연산을 수행



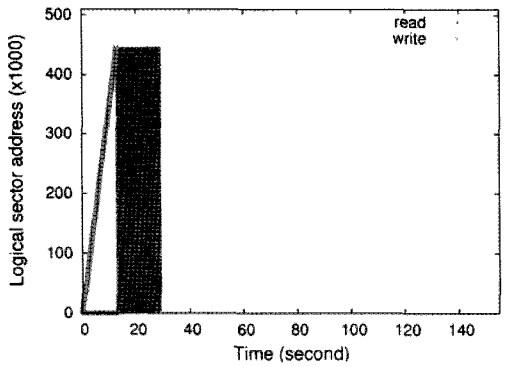
<그림 4> 데이터베이스 관리 시스템의 저장 장치 구조도

한다. 그런데, 만일 다수의 사용자가 동시에 데이터베이스에 대한 접근을 시도 할 경우, 임의 접근이 취약한 디스크의 특성으로 인해 빠른 응답을 제공하지 못하게 된다. 따라서 데이터베이스 관리 시스템은 빠른 응답속도를 위해 임의의 페이지들에 발생한 쓰기 연산을 "트랜잭션 로그"에 논리 물리적인 로그를 순차적으로 기록함으로써 빠른 응답속도를 제공하고 있다. 즉, 트랜잭션 로그 영역의 쓰기 패턴은 언제나 순차 쓰기로 이루어지며 이러한 특징은 앞서 설명한 SSD의 공통적인 특징과 일치한다. 또한 트랜잭션 로그의 경우, 데이터베이스 테이블스페이스에 비해 상대적으로 적은 용량을 필요로 한다. 따라서 트랜잭션 로그를 기록하는 저장매체로써 SSD를 사용할 경우, 상당한 성능 개선을 이룰 수 있다.

또한 데이터베이스 관리 시스템에서는 임시 데이터의 생성이 빈번하게 발생한다. 즉, 메모리의 용량에 비해 큰 데이터를 정렬할 경우, 색인 구조를 만들 경우 혹은 여러 개의 테이블을 조인 연산할 경우 임시로 생성되는 데이터를 기록하기 위한 영역이 요구되며 이 영역을 임시 테이블스페이스라 부른다. <그림 5>는 메모리의 크기가 2MB인 경우 200MB의 데이터에 대한 외부 정렬을 수행 한 결과로 임시 테이블스페이스를



(a) 하드 디스크의 외부 정렬



(b) SSD의 외부 정렬

〈그림 5〉 Temp. Table Space의 외부 정렬 비용

각각 하드 디스크와 SSD로 사용한 경우의 비용을 나타내고 있다. 그림과 같이 임시 테이블스페이스의 입출력 패턴은 순차쓰기와 임의 읽기의 형태로 나타나고 있으며 하드 디스크에 비해 SSD를 사용할 경우의 성능 향상이 월등히 뛰어남을 알 수 있다.

이처럼 SSD에 효과적인 입출력 패턴을 분석하여 데이터베이스 관리 시스템에 적용하는 연구의 경우 시스템의 별도의 변경 없이 적용이 가능하기 때문에 기업 환경의 데이터베이스 시스템에 손쉽게 적용할 수 있다.

최근 SSD 가격의 하락과 빠른 임의 쓰기 성능을 제공하는 SSD의 등장으로 인해 데이터베이스 테이블스페이스에 SSD를 적용할 수 있는 방법에 대한 연구도 최근 활발히 진행되고 있다. 앞서 설명한 것처럼 기존의 데이터베이스 관리 시스템은 로그 기법과 모듈별 디스크를 별도로 유지하는 등의 다양한 기법을 통해 성능 향상을 시켜왔으며, 데이터베이스 테이블스페이스의 경우 RAID를 통한 다수의 하드 디스크를 활용함으로써 데이터베이스의 성능을 향상시켜 왔다. 따라서 다수의 하드 디스크에 대한 설치비용과 다수의 하드 디스크를 유지하기 위한 전력 소비 비용이 발생하게 된다. 따라서 RAID 기반의 다수의 하드 디스크를 하나의 SSD로 대체하기 위한 연구가 진행 되고 있다.

<표 1>은 데이터베이스 관리 시스템의 대표적인 벤치마크인 TPC-C를 8개의 하드 디스크로 구성된 RAID-0과 하나의 SSD를 사용하는 데이터베이스를 구축하여 테스트한 실험 결과이다. 표에서 알 수 있듯이 전력 소비량의 편차가 항상 1.6배 이상이 발생함을 알 수 있다. 이에 반해 데이터베이스의 시간당 처리 횟수를 나타내는 TPS(transaction per second)의 차이가 거의 나타나지 않고 있다. 또한 CPU 사용률의 경우 SSD를 사용할 경우의 사용률이 큰 것을 알 수 있다. 즉, SSD를 사용할 경우 8개의 하드 디스크를 사용하는 시스템에 비해 보다 적은 전력 소비로 대등한 성능을 보임을 알 수 있다. 이러한 차이는 SSD의 성능이 비약적으로 발전하고 있

〈표 1〉 저장 장치 별 성능 및 전력 소비

Device	Idle Energy	Peak TPS (CPU util.)	Peak Energy
SSD	115.2 W	4269 TPS (50%)	141.0 W
Raid-0 (8 disks)	197.2 W	4274 TPS (45%)	231.4 W

는 점을 감안할 경우 더욱 크게 나타날 것으로 보인다.

V. 그 밖의 연구 동향 및 결론

지금까지 플래시메모리를 활용한 데이터베이스 연구 동향을 데이터베이스의 저장 관리자 위주로 소개 하였다. 이번 장에서는 저장 관리자 측면이 아닌 데이터베이스 분야에서 플래시메모리에 대한 연구가 어떻게 진행되고 있는 지에 대해 소개 하며 결론을 맺도록 한다.

1. 플래시메모리 기반의 색인 구조

플래시메모리를 저장 장치로 활용하기 시작하면서부터 색인구조(index)에 관한 연구는 비교적 빠르게 진행 되어 왔다. 색인구조에 대한 연구는 디스크 기반에서 다양하게 활용되고 있는 B^+-tree등과 같이 기존 색인 구조를 효과적으로 활용하기 위한 연구와 플래시메모리에 적합한 새로운 색인 구조에 대한 설계로 연구의 방향이 나뉜다. 또한 FTL을 기반으로 설계된 색인구조와 플래시메모리 인터페이스를 직접 활용하는 색인구조로 나누어 볼 수 있다. FTL을 기반으로 하는 연구들의 특징은 명확하다. 메모리 공간에서 최대한 임의쓰기를 순차쓰기의 형태로 바꾸기 위한 알고리즘을 활용하고 있다. 따라서 메모리의 사용량이 증가하고 결국 고장 회복(recovery)이 어려운 단점을 가지고 있다. 반면 플래시메모리 인터페이스를 직접 활용하는 색인구조들은 FTL에서 수행해야하는 소거 연산(garbage collection)과 논리적-물리적 주소 사상에 대한 추가 연산을 수행해야 하는 단점이 있다.

2. FTL

기존의 FTL에 대한 연구는 하드웨어 혹은 하드웨어 기반의 시스템 소프트웨어 연구 분야에서 집중적으로 진행되어 왔다. 그러나 최근에는 데이터베이스 분야에서 역시 FTL에 대한 연구가 활발히 진행되고 있다. 데이터베이스는 다른 응용 소프트웨어에 비해 주소 범위가 매우 크고 임의적인 특징이 있다. 또한 입출력의 단위가 크지 않기 때문에 기존의 대량의 순차쓰기에 강한 특징을 가지는 FTL을 사용할 경우 좋지 못한 성능을 보인다. 이러한 문제점을 해결하기 위해 데이터베이스 분야에서는 다양한 워크로드에 따라 입출력의 패턴을 분석하고 이에 적합한 FTL을 설계하는 연구가 진행되고 있다.

3. 기타 연구 동향

이상에서 본 바와 같이, 대용량 데이터를 취급하는 데이터베이스 분야는 플래시메모리 및 SSD의 가장 유망한 응용의 한 예로써 현재 다양한 분야의 연구가 진행 중이다. 우선 확장된 버퍼캐시의 일환으로 SSD를 사용하는 연구가 진행중인데, 이는 기존의 버퍼캐시의 교체정책과 결합해서 시스템 구축비용을 낮추고 성능을 향상시키게 될 것이다. 또한 데이터들의 접근밀도를 고려해서 데이터베이스의 데이터를 하드디스크와 SSD에 분산 배치하는 방법론에 대한 연구도 산업체를 중심으로 진행 중이다. 그리고 데이터베이스의 트랜잭션기술은 저장장치의 특성과 밀접한 관련이 있는데, 하드디스크와 다른 플래시메모리의 특성으로 인해, 기존 디스크 기반의 트랜잭션 기법과는 다른 새로운 트랜잭션 모델 및 구현기술 개발이 진행 중이다.

결론적으로 데이터베이스 분야는 플래시메모리의 가장 큰 응용분야의 하나이며, 또한 다양한 자료구조, 알고리즘, 복잡한 입출력에 대한 요구사항을 갖고 있기 때문에 지금까지 하드디스크 기반의 데이터베이스 기법들에 대해 플래시메모리 및 SSD 관점에서 새롭게 재조명해야 할 많은 기술적 내용들을 포함하고 있다.

참고문헌

[1] S. W. Lee, and B. Moon, "Design of Flash-Based DBMS: An In-Page Logging Approach," Int. Conf. on Management of Data (SIGMOD), 2007, pp.55-66.

[2] S. W. Lee, and B. Moon, et al. "A Case for Flash Memory SSD in Enterprise Database Applications," Int. Conf. on Management of Data (SIGMOD), 2008, pp.1075-1086.

[3] S. W. Lee, B. Moon, and C. Park. "Advances in flash memory SSD technology for enterprise database applications," Int. Conf. on Management of Data (SIGMOD), 2009, pp.863-870.

[4] Goetz Graefe. "The Five-minute Rule Twenty Years Later, and How Flash Memory Changes the Rules," In Third Int. Workshop on Data Management on New Hardware (DAMON2007), Beijing, China, June, 2007.

[5] Y. Li, B. He et al. "Tree Indexing on Flash Disks," In ICDE 2009.

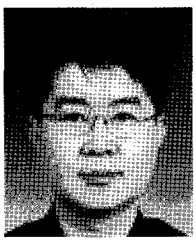
[6] Transaction Processing Performance Council. TPC Benchmark C Standard Specification (Revision 5. 9). http://www.tpc.org, June, 2007.

저자소개

나 갑 주

2003년 한국항공대학교 항공전자공학과(학사)
2006년 성균관대학교 컴퓨터공학과(석사)
2006년~현재 성균관대학교 전자전기컴퓨터공학과 박사과정

주관심 분야 : 데이터베이스, 플래시 메모리 응용, 색인 구조



이 상 원

1991년 서울대학교 컴퓨터학과(학사)
1994년 서울대학교 컴퓨터학과(석사)
1999년 서울대학교 컴퓨터학과(박사)
1999년~2001년 한국 오라클
2001년~2002년 이화여대 BK21 계약교수
2002년~현재 성균관대학교 정보통신공학부 부교수

주관심 분야 : 플래시 메모리 데이터베이스