

# 스토리지 클래스 메모리 사용을 위한 데이터베이스 워크로드 성능 특성 분석

이세호\*, 김정훈\*, 엄영익\*

\*성균관대학교 정보통신대학

e-mail: {loadic, myhuni20, yieom} @skku.edu

## Analyses of Database Workload for Storage Class Memory Systems

Seho Lee\*, Junghoon Kim\*, Yong Ik Eom\*

\*College of Information and Communication Engineering, Sungkyunkwan  
University

### 요 약

최근 연구 개발되고 있는 스토리지 클래스 메모리는 정체되어 있는 스토리지와 DRAM 산업에 큰 변화를 가져올 것으로 예상된다. 현재 컴퓨팅 환경에서 스토리지의 성능 저하요소가 큰 이슈로 야기되어 지는 가운데 본 논문에서는 TPC-C 벤치마크를 이용하여 임의 쓰기와 덮어 쓰기 연산 시 발생하는 문제점들을 분석한다. 실험 결과를 통해 향후 스토리지 클래스 메모리를 활용하여 기존 쓰기 연산 시 발생 하는 문제점들을 해결할 수 있는 방안에 대해 논의 한다.

### 1. 서론

최근 연구 개발되어지고 있는 차세대 스토리지 클래스 메모리(Storage Class Memory)는 정체되어 있는 스토리지와 DRAM 산업에서 주목 받고 있는 기술 중의 하나이다. 이는 스마트 디바이스, 클라우드 서버, 데스크톱 등 많은 분야에 적용되어 차세대 컴퓨팅 환경의 변화를 주도 할 수 있는 장치이다. 현재 이를 활용한 많은 연구들이 진행되고 있다[1][3].

이전 연구 결과들에 따르면 현재 컴퓨팅 환경에서의 성능 향상을 가져 올 수 있는 것은 프로세서와 네트워크 성능이 아닌 스토리지라고 야기되고 있다[2]. 특히, 대부분의 유명 어플리케이션 (facebook, firefox, chrome, skype)등에서 SQLite[4]가 사용되어 지고 있다. 이것은 빈번한 임의 쓰기과 덮어 쓰기(Overwrite)를 발생시켜 빈번한 디스크 접근을 유발하여 컴퓨팅 성능 저하를 가져오는 요인으로 지적되고 있다. 따라서 이를 해결하기 위해 현재 스토리지 관점에서의 임의 쓰기과 중복 쓰기 오버헤드를 최소화 하려는 많은 연구들이 진행되고 있다. 그러나 현재의 컴퓨팅 환경에서는 이를 해결할 방법이 부족한 상황이다.

본 논문에서는 현재 컴퓨팅 환경의 문제점을 해결 하기 위한 방안을 모색하기위해 현실적인 응용 프로그램들의 환경들을 본 따 설계되어 있는 TPC-C 벤치마크를 수행하여 실제 임의 쓰기과 중복 쓰기의 문제점을 분석한다. 또한 이러한 문제점을 차세대 메모리를 활용하여 좀 더

효과적으로 해결할 수 있는 방안에 대해서 논의한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 배경 지식에 대해 설명하고, 3장에서는 실험 평가 및 분석을 통해 쓰기 연산 시 문제점을 분석한 뒤 해결 방안에 대해 논의 한다. 4장에서는 본 논문에 대한 결론 및 향후 연구 방향을 제시한다.

### 2. 배경 지식

#### 2.1. 스토리지 클래스 메모리

스토리지 클래스 메모리는 비 휘발성 메모리로 최근 PCRAM, STT-MRAM 등 다양한 형태로 개발되고 있다. 스토리지 클래스 메모리는 비 휘발성 메모리라는 특징 이외에도 기존 DRAM처럼 바이트 단위로 접근 가능하며, 낸드플래시 메모리 이상의 접근 속도를 제공하는 특징을 가지고 있다. 이는 향후 기존 DRAM 혹은 디스크를 대체 하거나 새로운 컴퓨터 구조를 가져올 것으로 예상된다.

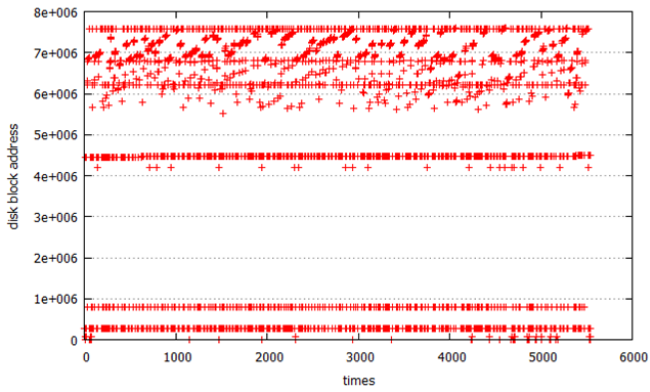
#### 2.2. TPC-C

TPC[5](Transaction Processing Performance Council)는 비영리 시스템 성능 테스트 평가기관이다. TPC-C 벤치마크는 온라인 거래처 처리(OLTP) 벤치마크로 하드웨어나 OS에 상관없이 어느 트랜잭션 처리 시스템 상에서 실행될 수 있는 일련의 기능 요구사항을 정의하기 위해 구현되었다. 이는 터미널 사용자들의 집합이 데이터베이스에 대한 거래를 실행하는 환경을 시뮬레이션 한다. 본 논문에서는 이러한 환경을 제공하는 TPC-C 벤치마크를 수행하여 겹쳐 쓰기 되는 워크로드 특성에 대해 분석한다.

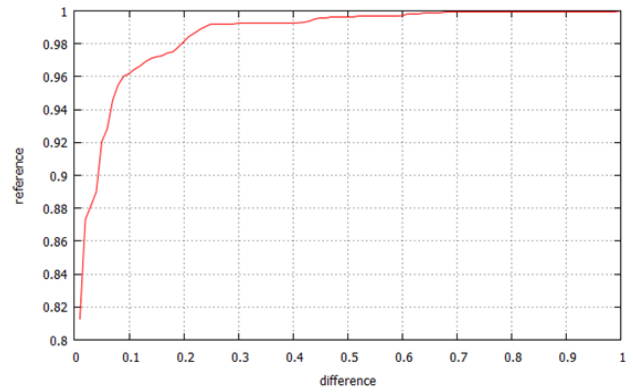
### 3. 실험 평가 및 분석

본 장에서는 TPC-C 벤치마크를 이용하여 쓰기 연산

† 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041244, 스마트TV 2.0 소프트웨어 플랫폼]



(그림 2) 디스크 블록 주소 별 쓰기 패턴



(그림 1) 덮어 쓰기 연산 시 실제 변경된 디스크 블록의 크기

시 발생하는 워크로드 특성을 실험 평가 및 분석한다.

### 3.1. 실험 환경

실험 환경은 QEMU와 KVM 기반의 가상머신을 이용하여 SQLite를 기반으로 TPC-C 벤치마크를 수행 하였다. 덮어 쓰기 발생 시 디스크 블록에 기록되는 데이터를 캡처하여, 덮어 쓰기 전 디스크 블록과 1 Byte 단위로 비교하여 얼마나 많은 부분이 실제로 변경되어 디스크에 기록되는지 실험하였고, 어떤 패턴으로 디스크 블록 주소에 접근하는지 기록하였다.

### 3.2. 실험 결과 및 분석

그림 1은 쓰기 연산 시 디스크 블록 주소별 접근 패턴을 나타낸다. TPC-C 벤치마크에서 쓰기 발생 시 순차적으로 디스크에 접근 하는 것이 아닌, 편향된 임의 쓰기형태로 디스크에 접근하는 것을 확일 할 수 있다. 현재 디스크를 사용하는 컴퓨팅 환경에서는 이전 연구[2]에서 보여주는 것처럼 임의 쓰기 연산은 순차적으로 디스크에 접근 하는 것에 비해 많은 오버헤드를 가지고 있어 성능 저하 요소로 지적 되고 있다.

위의 그림 2는 덮어 쓰기 발생 시 실제로 디스크 블록에 저장되는 데이터와 이전에 디스크에 저장된 디스크 블록과의 차이에 대한 누적 분포를 보여준다. TPC-C 벤치마크를 수행하는 경우 실제로 덮어 쓰기 발생 시 변경된 데이터는 4 Kbyte 블록 중 대부분 10%의 데이터만 변경된 블록이 디스크로 저장 된다. 결국 4 Kbyte 블록 중 90% 이상의 동일한 데이터가 다시 디스크로 써진다. 이는 실제로 임의 쓰기와 더불어 불필요한 디스크 접근 유발하여 현재 데이터베이스를 사용하는 컴퓨터의 성능 저하를 가져온다.

현재 연구 개발되고 있는 바이트 단위 접근 가능한 스토리지 클래스 메모리와 기존의 디스크를 함께 사용[1]하여 불필요하게 디스크에 접근되는 횟수를 줄이거나, 기술 발전으로 인해 기존의 디스크를 대용량의 스토리지 클래스 메모리로 대체한다면[3], 현재 임의 쓰기 연산이 빈번하게 발생되어 상대적으로 느린 디스크 접근으로 인해 발생하는 성능 저하 현상을 제거 할 수 있을 것이다. 이러한 파일 시스템을 디자인 하고 설계한다면, 실제 컴퓨팅 성능

향상에 도움을 줄 것으로 기대된다.

## 4. 결론 및 향후 연구 방향

본 논문에서는 TPC-C 벤치마크를 이용하여 쓰기 연산 시 발생하는 디스크 접근 패턴과 덮어 쓰기 연산 시 실제 변경된 데이터의 양을 측정하였다. 측정된 결과는 임의 쓰기 연산으로 발생하는 성능 저하 문제와 실제 덮어 쓰기 연산 시 기존 데이터 대비 변경되지 않은 데이터가 90% 이상임을 확인 할 수 있었다. 이는 현재 연구 진행 중인 스토리지 클래스 메모리를 활용한다면 본 연구에서 나타난 문제점을 해결할 수 있을 것으로 기대된다. 따라서 향후에는 스토리지 클래스 메모리를 활용하여 본 논문에서 보여준 쓰기 연산 시 발생하는 문제점들에 대한 해결 방안을 파일 시스템 관점에서 연구할 계획이다.

### 참고문헌

- [1] Eunji Lee, Hyokyung Bahn, and Sam H. Noh, "Unioning of the Buffer Cache and Journaling Layers with Non-volatile Memory," Proceedings of the 11th USENIX conference on File and Storage Technologies, p.73-80, February 12-15, 2013, San Jose, CA
- [2] Hyojun Kim, Nitin Agrawal, and Cristian Ungureanu, "Revisiting Storage for Smartphones," Proceedings of the 10th USENIX conference on File and Storage Technologies, p.209-222, February 14-17, 2012, San Jose, CA
- [3] Jeremy Condit, Edmund B. Nightingale, Christopher Frost, Engin Ipek, Doug Burger, Benjamin Lee, and Derrick Coetzee, "Better I/O Through Byte-Addressable, Persistent Memory," Proceedings of the 22nd Symposium on Operating Systems Principles (SOSP 2009), October 11-14, 2009, Big Sky, MT
- [4] SQLite, <http://www.sqlite.org/>
- [5] Transaction Processing Performance Council, <http://www.tpc.org/>