OpenSSD 플랫폼에서 선택적 캐싱 기반 페이지 매핑 테이블의 캐시 비율에 따른 성능 모델 연구

강동현[1], 하영남[2], 임희락[2], 김영재[1]
[1]서강대학교 컴퓨터공학과, [2]아주대학교 소프트웨어학과
{augustkang, youkim}@sogang.ac.kr, {haha1357, sgo1212}@ajou.ac.kr

Performance Modeling and Measurement of Selective Page-Mapping Table on the OpenSSD Platform

Donghyun Kang[1], Youngnam Ha[2], Heerak Lim[2], Youngjae Kim[1]
[1]Sogang University, [2]Ajou University

요 약

SSD에서 Flash Translation Layer (FTL)은 논리적인 주소를 물리적으로 주소로 바꾸어 주는 역할을 수행한다. FTL은 SSD의 성능에 크게 영향을 미친다. 선택적 캐싱 기반 페이지 매핑 FTL은 자주 접근되는 페이지 매핑 엔트리만 메모리에 유지하며 페이지 매핑 테이블을 구현하는 방식이며 Garbage Collection 오버헤드를 최소화하여 SSD 성능을 높인다. 하지만, 워크로드의 특성상 Locality가 낮은 경우 캐시 미스가 발생하여 성능이 낮아지는 문제점을 가진다. 본 논문에서는 이러한 선택적 캐싱 기반 페이지 매핑 FTL의 캐시된 매핑 테이블 크기에 따라 성능 분석을 수행한다. 특히 성능 예측 모델 개발을 위해 OpenSSD 플랫폼 상에서 선택적 캐싱 기반 페이지 매핑 FTL을 실제 구현하였으며 파일 시스템 성능 분석 도구를 이용하여 FTL 성능 예측 모델의 정확성을 검증하였다.

1. 서론

최근 컴퓨터의 저장장치로 SSD의 사용이 증대되고 있다. SSD의 성능은 내부에 위치한 FTL (Flash Translation Layer)에 큰 영향을 받는다. FTL의 주요 기능 중 Address translation은 물리적 페이지와 논리적 페이지를 사상하는 기능이다. Address translation 방식에는 Block-level 맵핑 방식, Page-level 맵핑 방식, 그리고 Hybrid 맵핑 방식이 있다. Hybrid 맵핑 방식은 Page level, Block level 의 장점을 혼용한 방식으로, Garbage collection이 발생할 때 Full merge로 인한 시스템 전체 성능의 저하를 발생 시킨다 [1].

Demand-based Page-mapped FTL (DFTL)은 Hybrid FTL이 가진 단점을 해결할 수 있도록 제안된 선택적 캐싱 기반 Page-level FTL이다 [1]. DFTL은 Cached Mapping Table (CMT)과 Global Translation Directory (GTD)를 이용하여, 엔터프라이즈 워크로드에 대해 좋은 성능을 보여준다. 현재 DFTL 연구는 시뮬레이터를 활용한 연구에만 제한되어 있다. DFTL의 성능은 CMT 크기와 Page-level FTL을 DRAM에서 캐싱하는 비율에 크게 영향을 받는다. 실제 DRAM 캐시 Hit Ratio에 따른 성능 변화를 확인해보기 위해 본 연구에서는 OpenSSD 플랫폼에서 DFTL을 구현하고 성능 모델을 개발하였다. 특히 본 연구에서는 선형회귀 기법을 사용하여 DFTL의 성능 모델 예측이 가능하다는 것을 밝힌다.

2. Open SSD 기반 DFTL 설계 및 구현

본 연구에서는 Page-level FTL의 캐싱 비율에 따른 Address translation 성능을 실제 SSD에서 실험하고자 DFTL을 직접 구현하였다. 코드 개발은 OpenSSD Jasmine Board를 사용하였다 [2].

DFTL에서는 SRAM을 GTD와 CMT를 위한 공간으로 사용하는 것으로 제안하였지만, 본 논문에서 사용한 보드는 SRAM size가 전체 플래시 메모리 저장 공간에 상응하는 페이지 매핑 테이블을 수용할 정도로 충분히 크지 않기 때문에 DRAM 영역을 이용하였다. 플래시 메모리 영역에 전체 페이지 매핑 테이블을 저장하도록 하고 SSD에 전원이 인가될 때 플래시 메모리에 저장되어 있는 전체 저장공간에 대한 Page-level FTL을 DRAM으로 적재 하도록 하였다.

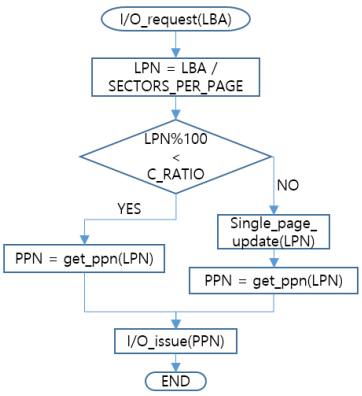

Figure 1 - I/O 순서도

Figure 1은 DFTL이

Read/Write 요청을 실행할 때 I/O 처리 순서도를 보여준다. 호스트로부터 주어진 논리적 블록 주소 LBA로 I/O 요청을 받으면 LBA를 페이지당 섹터 수 (SECTORS_PER_PAGE) 로 나눈 몫을 LPN 값으로 계산한다. 저장공간의 기본 단위인 섹터는 512B로써, 페이지 크기를 512B로 나눈 값이 페이지당 섹터 수이다. 하나의 블록은 128개의 페이지로 이루어져 있는데, 논리적 블록 주소를 페이지 당 섹터 수로 나눈 몫으로 I/O를 요청할 논리적 페이지 번호를 알 수 있다.

DFTL은 실험자가 캐싱 비율 (C_RATIO, %)을 임의로 정할 수 있다. 실험자가 정한 C_RATIO 만큼의 요청은 DRAM의 Page-level FTL을 참조하여 물리적 페이지 번호를 구하고, 나머지 (100 - C_RATIO) 만큼의 요청은 플래시 메모리에 저장된 Page-level FTL을 참조해 물리적 페이지 번호를 구한다. Figure 1의 조건 분기(LPN%100 < C_RATIO) 는 이를 나타낸다.

DFTL에서 플래시 메모리의 Page-level FTL을 참조는 CMT에서 Miss가 발생을 말한다. 그러므로, LPN을 100으로 나눈 나머지가 C_RATIO보다 작은 경우 Miss가 발생한 CMT의 엔트리를 업데이트 시키는 과정이 수반된다. single_page_update(LPN) 함수는 이러한 역할을 하는 함수로써, 플래시 메모리에 저장된 Page-level FTL에서 LPN에 상응하는 PPN을 엔트리로 담고 있는 페이지를 찾아 DRAM의 캐시된 Page-level FTL 엔트리를 업데이트 시킨다. get_ppn(LPN) 함수는 주어진 LPN에 해당하는 PPN을 DRAM의 캐시된 Page-level FTL에서 읽어오는 함수이다. 최종적으로 PPN을 구한 이후 FTL은 해당 PPN에 요청 받은 I/O를 실행한다.

3. 성능 모델

본 논문에서는 선형회귀 분석 방법을 사용하여 CMT 캐싱 비율에 따른 성능 예측 모델을 제시하고 OpenSSD 플랫폼에서 이를 검증한다. 기존 관련 연구들은 모델을 수립하고 시뮬레이터 또는 특정 SSD 제품들에 대해 모델 검증을 진행하였으나, 회귀 모델을 바탕으로 DFTL의 캐싱 비율에 따른 성능을 확인하는 연구는 다루지 않았다.[3,4] 본 논문의 모델을 위한 회귀분석은 다중 선형 회귀 분석 방법(Multiple regression analysis), 단순 회귀 분석 방법(Linear regression analysis)을 사용한다. 다중 회귀 분석은 실제 실험을 통해서 구한 Page-level FTL 캐싱 비율과 IOPs의 관계, Read/Write 비율과 IOPs의 관계를 Exponential 모델을 기반으로 분석한다. 이 회귀분석을 기반으로 실험을 통해 구하지 않은 DTFL의 성능이나 필요한 캐싱 비율을 예측할 수 있다.

(i) 다중 회귀 분석 모델

$$\log(\text{IOps}) = \hat{\alpha} + \widehat{\beta_1} * (Dram\ \ Caching\ \ Ratio) + \widehat{\beta_2} * (Read\ \ Ratio) + e$$

(ii) 선형 회귀 분석 모델

$$\log(\text{IOps}) = \hat{\alpha} + \hat{\beta} * (Dram\ \ Caching\ \ Ratio) + e$$

4. 실험 및 검증

구현한 DFTL을 테스트 환경은 Table 1과 같다.

	CPU	Memory
Host PC	Intel i7-3770 3.4GHz	4GB DRAM
Jasmine board[2]	ARM7TDMI-S 87.5MHz	96KB SRAM 64MB SDRAM

Table 1 – 테스트 환경

대표적인 엔터프라이즈 워크로드를 사용하여 실험을 진행했으며, 워크로드의 특성은 Table 2와 같다.

워크로드	Avg.Req Size(KB)	Read (%)	Seq (%)	Avg. Req. Inter-arrival Time (ms)
Financial [5]	4.38	9.0	2.0	133.50
TPC-H [6]	12.82	95.0	18.0	155.56

Table 2 – 사용 워크로드 특성

Figure 2는 두 워크로드의 캐싱 비율에 따른 평균 I/O 응답 처리 시간(Average Response Time)을 나타낸다. Figure 3은 캐싱 비율에 따른 IOPs를 나타낸다.

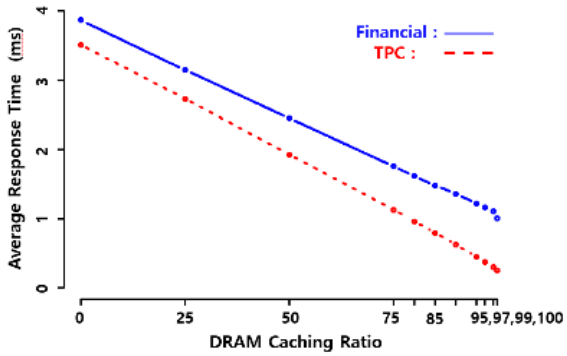

Figure 2 – Average response time과 캐싱 비율 관계

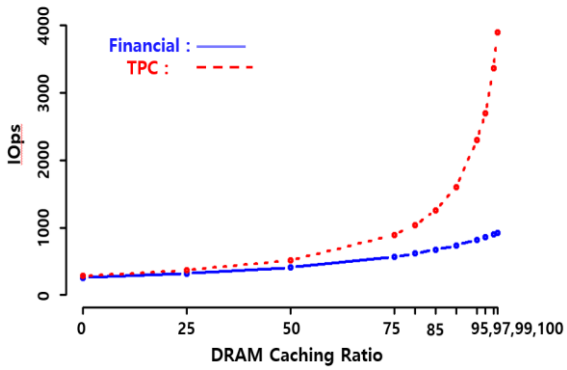

Figure 3 – IOPs와 캐싱 비율 관계

두 워크로드는 Request size와 Read / Write 비율, Sequential access의 비율이 서로 다르다. SSD는 Sequential/Random access의 차이가 크지 않기 때문에, 다중 회귀분석 고려 대상에서 배제하였다. Request size가 다중 회귀 분석 모델에 주는 영향을 확인하기 위해 Request size를 변경하며 단순 회귀분석의

결정계수(R^2)값 변화를 관찰했다. 실험 결과, Request size를 1/3 감소시킨 워크로드에서도 결정계수(R^2)값은 약 1.15% 정도의 작은 변동만 있음을 확인했다. 따라서 Request size도 회귀분석에 큰 영향을 미치지 않음을 알 수 있다. 마지막으로, Read / Write 비율에 따른 결정계수(R^2)값 변화도 확인했다. 실험 결과 결정계수(R^2)값은 전반적으로 반비례의 성격을 나타냈고, Read 비율이 100에 가까워질수록 결정계수(R^2) 값의 감소 폭이 더 큰 것으로 나타났다.

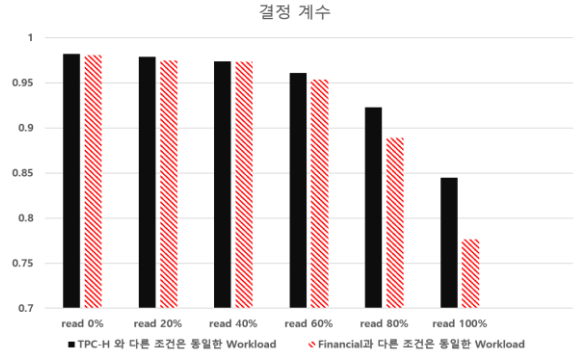

Figure 4 – Read/Write 비율에 따른 결정계수의 변화

결과를 바탕으로 다중 회귀 분석과 단순 회귀분석의 수학적 모델을 워크로드 별로 다음과 같이 수립하였다.

Financial	$\log(\text{IOPs}) = 4.588 + 0.012 * (Dram\ \ Caching\ \ Ratio) + 0.021 * (Read\ \ Ratio)$
TPC-H	$\log(\text{IOPs}) = 4.629 + 0.012 * (Dram\ \ Caching\ \ Ratio) + 0.019 * (Read\ \ Ratio)$

Table 3 – 다중 회귀 분석 모델

Financial	$\log(\text{IOPs}) = 5.456 + 0.013 * (Dram\ \ Caching\ \ Ratio)$
TPC-H	$\log(\text{IOPs}) = 5.287 + 0.025 * (Dram\ \ Caching\ \ Ratio)$

Table 4 – 단순 회귀 분석 모델

위 선형 회귀 모델의 F-test p-value는 Financial 워크로드의 경우 $1.358 * 10^{-7}$ 이고 TPC-H 워크로드의 경우 $2.198 * 10^{-5}$ 이다. 즉 위 모델에서 IOPs와 DRAM 캐싱 비율이 선형적 관계임을 99.9%의 신뢰도로 나타낸다. 각 워크로드에 대한 회귀곡선 모델의 결정계수(R^2)값은 Financial의 경우 0.974, TPC-H의 경우 0.877이다. 이는 워크로드에 대한 데이터가 회귀곡선 모델로서 설명될 수 있는 부분이 각각 97.4%, 87.7%임을 의미한다.

Figure 5,6은 Table 4의 단순회귀 분석 모델을 나타낸 그래프로써, 각각의 워크로드에서 예측되는 IOPs와 Page-level FTL 캐싱 비율의 관계를 보인다. 두 워크로드를 적용했을 때 목표하는 IOPs를 기준으로 얼마만큼의 Page-level FTL을 DRAM에서 캐싱해야 하는지 다음과 같이 예측해 볼 수 있다.

예를 들어, Financial 워크로드에서 700 IOPs의 성능을 내고자 할 경우 CMT의 캐싱 비율이 최소 84%는 되어야 함을 예측할 수 있다. 마찬가지로 Figure 6의 TPC-H 워크로드에서 1500 IOPs의 성능을 내고자 할 경우 CMT 캐싱 비율이 최소 81%는 되어야 함을 예측할 수 있다.

5. 결 론

본 연구를 통해 DFTL의 캐싱 비율에 대한 FTL 성능을 OpenSSD Platform상에서 검증하였다. DFTL 논문에서 제시된 캐싱 비율을 실제 SSD 보드에 적용할 경우, 시뮬레이터 상에서 측정되었던 성능보다 저하된 성능을 보이는 것을 실험으로 확인하였다. 실험 결과를 바탕으로 개발한 성능 예측 모델을 통해 특정 워크로드의 Locality에 따른 수용 가능한 DRAM 캐싱 비율을 예측할 수 있었다.

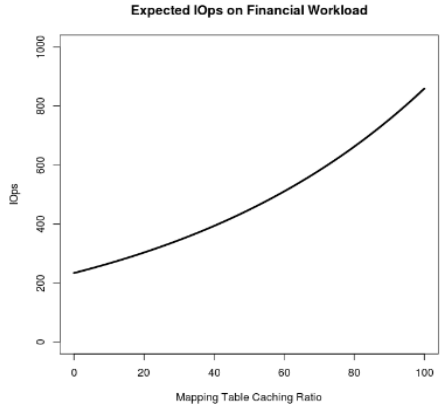

Figure 5 – Financial IOPs에 따른 캐싱 비율 예측

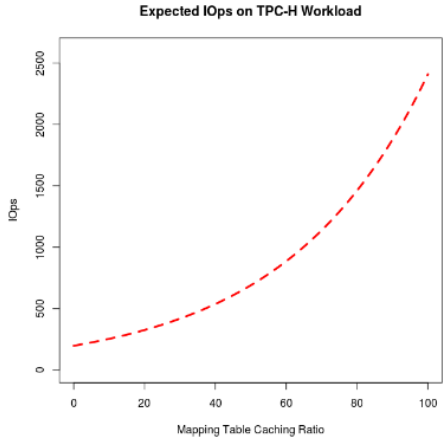

Figure 6 – TPC-H IOPs에 따른 캐싱 비율 예측

ACKNOWLEDGEMENT

본 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원 (No. R0190-15-2012)과 한국 연구재단의 지원 (No. 2015R1C1A1A0152105), 그리고 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학 지원사업의 연구결과로 수행되었음.

6. 참고 문헌

[1] A. Gupta, Y. Kim, B. Urgaonkar. *DFTL: a flash translation layer employing demand-based selective caching of page-level address mappings.* ASPLOS 2009.
[2] OpenSSD Project, 2016, http://www.openssd-project.org.
[3] Boboila S, Desnoyers P. *Performance models of flash-based solid-state drives for real workloads.* IEEE 27th MSST 2011.
[4] Huang, H. Howie, et al. *Performance modeling and analysis of flash-based storage devices.* IEEE 27th MSST 2011.
[5] OLTP Trace from UMass Trace Repository. OLTP Trace from UMass Trace Repository. http://traces.cs.umass.edu/index.php/Storage/Storage.
[6] TPC BENCHMARKTM H (Decision Support) Standard Specification Revision 2.17.1, TPC-H benchmark, http://www.tpc.org/tpc_documents_current_versions/current_specifications.asp