

# SSD 성능에 영향을 주는 특징 분석 방법

강운석, 조용연, 김상욱<sup>1</sup>

한양대학교

e-mail : {poyche, jyy0430, wook}@hanyang.ac.kr

## A Method for Analyzing Features that Affect the Performance of SSD

Yoonsuk Kang, Yong-Yeon Jo, and Sang-Wook Kim  
Hanyang University

### 요 약

본 논문에서는 SSD 성능에 영향을 주는 특징(또는 특징 집합)을 평가하는 방법을 제안한다. 제안하는 평가 방법은 기존 연구의 “두 응용프로그램에서 추출한 IO 트레이스들이 서로 유사한 IO 패턴을 갖을 때, 동일한 SSD 에서의 수행 시간은 유사하다.”는 관찰에 기반한다. 이를 통하여 우리는 주어진 SSD 에서 후보 특징들을 평가하고, 가장 높은 평가를 받는 특징 집합을 확인한다.

### 1. 서론

기존 연구에서 “두 응용프로그램에서 추출한 IO 트레이스들이 서로 유사한 IO 패턴을 갖을 때, 동일한 SSD 에서의 수행 시간도 유사하다.”는 관찰이 있다[1]. 이 관찰을 이용하면 수행 시간을 알고자 하는 쿼리 IO 트레이스( $Q$ )의 수행시간을 다음과 같이 예측할 수 있다.  $Q$ 의 IO 패턴과 유사한 또다른 IO 트레이스( $S$ )가 존재하고,  $S$ 의 수행시간은  $t_s$ 라 가정하면,  $Q$ 의 수행시간  $t_q$ 은  $t_s$ 와 유사할 것으로 추정할 수 있다.

만약 타깃 SSD 에서 타깃 응용프로그램의 수행시간을 이와 같이 예측할 수 있다면, 가장 빠른 SSD 를 구매하고자 하는 소비자와 가장 좋은 성능의 SSD 를 제조하고자 하는 제조사에게 매우 큰 도움을 줄 수 있다. 타깃 SSD 에서 가장 빠른 응용프로그램이 무엇인지 쉽게 알 수 있고, 타깃 응용프로그램에서 가장 빠른 SSD 가 무엇인지 알 수 있기 때문이다.

본 논문에서는 기존 연구 관찰을 잘 만족하는 특징들이 어떤 것들이 있는지 확인한다. 이를 위해, 각 후보 특징 (또는 후보 특징들을 조합한 특징 집합들)이 기존 연구 관찰을 얼마나 만족하는지 평가하는 함수를 제안한다. 제안하는 함수를 통하여 각 후보 특징과 후보 특징 집합들을 평가한다.

### 2. 후보 특징 선정

우리는 위의 관찰을 만족하는 특징(또는 특징 집합)을 찾기 위해, 먼저 SSD 성능에 영향을 주는 후보 특징을 찾는다. SSD 성능에 영향을 주는 특징들 중에서 실제로 위의 관찰을 만족시키는 특징들이 존재할 가능성이 있기 때문이다.

<표 1>은 SSD 성능에 영향을 주는 요소들을 나타낸다. IO 의 수, IO 요청 크기, 연속된 두 IO 의 요청 시간 간격, IO 들이 참조하고 있는 논리주소 (LBA), 읽기 및 쓰기 IO request 개수 비율과 같은 요인들이

SSD 의 성능에 영향을 준다고 알려져 있다 [2][3].

<표 1> 후보 특징 리스트

Candidate feature ID	Candidate features	Meanings
1	IO count	of IOs in an IO trace
2	IO size(avg)	Average IO size in an IO trace
3	IO size(std)	Standard deviation of IO size in an IO trace
4	IO interval(avg)	Average request time interval between two adjacent IOs in an IO trace
5	IO interval(std)	Standard deviation of request time interval between two adjacent IOs in an IO trace
6	IO locality(avg)	Average LBA of IOs in an IO trace
7	IO locality(std)	Standard deviation of LBA of IOs in an IO trace
8	Read ratio	Ratio of read requests in an IO trace
9	Write ratio	Ratio of write requests in an IO trace
10	Sequential read ratio	Ratio of sequential read requests in an IO trace
11	Sequential write ratio	Ratio of sequential write requests in an IO trace
12	Random read ratio	Ratio of random read requests in an IO trace
13	Random write ratio	Ratio of random write requests in an IO trace

### 3. 후보 특징 평가 방법

본 논문에서는 후보 특징(또는 후보 특징 집합)이 기존 연구 관찰을 얼마나 만족하는지 평가하는 goodness function(GF)을 제안한다. GF 는 주어진 특징 (또는 특징 집합)을 기준으로 각 IO 트레이스로부터 특징 벡터를 추출했을 때, 특징 벡터가 유사한 IO 트레이스들의 수행시간이 얼마나 유사한지 정량적으로 평가하는 함수이다.

아래의 식과 같이 DB 내의 각 IO 트레이스  $w_j$  와 특징 집합  $F_i$  를 기준으로 이와 유사한  $p$  개의 IO 트레이스들( $w_k | w_k \in SW_j$ )에 대해서,  $w_j$ 와 각  $w_k$ 의 수행시간 ( $tw_j - tw_k$ ) 차이의 평균을 측정하고, 이를 모든 IO 트레이스에 대해서 반복한 평균 값을 GFscore 의 결과 값으로 한다.

GFscore 가 높을수록 유사한 IO 트레이스들의 수행 시간 차이가 작다는 것을 의미하고, 이는 곧  $F_i$  를 기

<sup>1</sup> 교신저자(wook@hanyang.ac.kr)

준으로 찾은 유사한 IO 트레이스들의 수행시간도 유사하다는 것을 의미한다.

$$GFscore(F_i) = \frac{1}{|W| \sum_{w_j \in W} \frac{1}{|P|} \sum_k^P |t_{w_j} - t_{w_k}|}$$

- $F_i$ : 특징 집합  $F_i$
- $w_j$ : IO 트레이스  $DB(W)$ 내의 타깃 DB IO 트레이스
- $w_k$ :  $w_j$ 와  $F_i$ 를 기준으로 유사한 DB IO 트레이스
- $t_{w_j}$ :  $w_j$ 의 수행시간
- $t_{w_k}$ :  $w_k$ 의 수행시간

#### 4. 특징 평가 실험

##### 4.1. 실험 환경

본 논문에서는 인텔 i7-920 2.67GHz, 12GB main memory, Linux 3.5.0-23 에서 실험을 수행하였다. 실험에 사용된 SSD 는 하이닉스 SSD 로 읽기 속도 510.5MB/s, 쓰기 속도 357.9MB/s 이다. 본 실험에서는 TPC-C, uFLIP, websearch IO 트레이스를 사용하였다. <표 2> 는 실험에서 사용된 전체 IO 트레이스에 대한 정보를 나타낸다.

<표 2> 사용된 데이터 셋

	# of IO traces	Total size of IO requests (GB)
TPC-C	18	143
uFLIP	486	419.1
Websearch	3	3.9
Total	507	566

##### 4.2. 실험 결과

우리는 가장 먼저 각 후보 특징이 얼마나 기존 연구 관찰을 만족하는지 GF 로 평가하였다. (그림 1)a 는 각 후보 특징을 GF 로 평가한 결과를 나타낸다. x 축은 후보 특징 ID 를 나타내고, y 축은 GFscore 를 나타낸다. 2 번 후보 특징(IO size(avg))이 가장 좋은 평가 결과를 보였고, 10 번 후보 특징(sequential read ratio)이 가장 안좋은 평가 결과를 보였다.

IO 트레이스 내의 각 IO 요청의 크기가 SSD 의 성능에 영향을 많이 주기 때문에 2 번 후보 특징이 가장 좋은 평가를 받은 것으로 보인다. 반면, SSD 는 순차적 읽기를 매우 빠르게 처리함에도 불구하고, 실제 응용프로그램 내에는 순차적 읽기의 비율이 매우 적기 때문에 평가 결과 가장 낮게 나온 것으로 보인다. 다음으로 우리는 후보 특징을 조합한 특징 집합들 중에서 가장 평가 결과가 좋은 후보 특징이 무엇인지 확인해보았다. 특징의 개수가 13 개 일 때, 모든 가능한 특징 집합의 수는  $2^{13}$  이다. 이를 일일이 GF 로 평가하는 것은 NP-hard 문제로 실질적으로 수행이 불가능하다<sup>2</sup>.

우리는 이를 해결하기 위하여 greedy approach 를 이용한다. 후보 특징의 개수가  $f$  개인 후보 특징 집합(CF) 이 주어졌을 때, greedy approach 는 특징 집합 내의 특징의 개수가 1 개일때부터 시작해서 특징을 하나씩 추가하면서  $f$  개일 때까지의 특징 집합 조합한다. 이때,  $(f-1)$ 개로 구성된 특징 집합과 이 특징 집합에 포함되지 않은 후보 특징 들 중에서 가장 높은 평가 결과를 받은 후보 특징이  $f$  개의 특징으로 구성된 특징

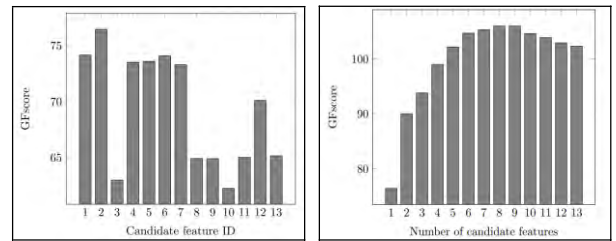
집합에 추가된다.

<표 3>은 greedy approach 방식을 이용하여 특징 집합에 추가된 순서와 각 후보 특징을 평가했을 때의 순위를 나타낸다. 2, 9, 7, 4, 11, 10, 3, 1, 5, 6, 12, 13, 8 의 순서로 특징 집합에 추가되었는데, 특징 집합에 추가되는 후보 특징의 순서가 각 후보 특징을 평가했을 때의 순위와 매우 다른 결과를 보였다.

<표 3> 특징 집합에 추가된 후보 특징 순서

Greedy approach rank	2	9	7	4	11	10	3	1	5	6	12	13	8
Candidate feature rank	2	1	6	5	4	7	12	13	11	8	9	3	10

(그림 1)b 는 특징 집합 내 특징 개수에 따른 GFscore 를 나타낸다. x 축은 특징 집합내 특징의 개수를 나타내고, y 축은 GFscore 를 나타낸다. 특징 8 개를 사용했을 때, 가장 높은 결과를 나타냈다.



a. 각 후보 특징                      b. 특징 집합

(그림 1) 평가 결과

#### 5. 결론

본 논문에서는 SSD 성능에 영향을 주는 특징(또는 특징 집합)을 평가하는 방법을 제안하였다. 제안하는 평가 방법은 기존 연구의 “두 응용프로그램에서 추출한 IO 트레이스들이 서로 유사한 IO 패턴을 갖을 때, 동일한 SSD 에서의 수행시간은 유사하다.”는 관찰을 얼마나 만족시키는지 평가한다. 이를 통하여 우리는 주어진 SSD 에서 후보 특징들을 평가하고, 기존 연구 관찰을 잘 만족하는 특징 집합을 확인하였다. 향후 연구로는 제안하는 방법보다 더 효율적으로 특징(또는 특징 집합)을 평가하는 방법과 해당 평가 방법을 이용하여 얻은 특징 집합을 이용하여 타깃 응용프로그램의 수행시간을 예측하는 연구를 진행할 예정이다.

#### 감사의 글

이 논문은 (1) 한국연구재단-차세대정보·컴퓨팅기술 개발사업(No. NRF-2017M3C4A7069440), (2) 한국연구재단 (No. NRF-2017R1A2B3004581), 그리고 (3) 과학기술 정보통신부 및 정보통신기술진흥센터의 대학 ICT 연구 센터육성 지원사업(IITP-2018-2013-1-00881)의 지원을 받아 수행된 연구임.

#### 참고문헌

[1] H. Huang et al., “Performance Modeling and Analysis of Flash-Based Storage Devices,” In *Proc. IEEE MSST*, pp. 1-11, 2011.

[2] S. Lee et al., “IO Workload Characterization Revisited: A Data-Mining Approach,” *IEEE Trans. on Computers*, Vol. 63, No. 12, pp. 3026-3038, 2014.

[3] L. Bouganim, B. J ’onsson, and P. Bonnet, “uFLIP: Understanding Flash IO Patterns,” In *Proc. CIDR*, 2009.

<sup>2</sup> 현재 데이터에서 모든 특징 집합을 평가하는데 걸리는 시간은 48000 시간정도 걸릴 것으로 예상됨