

나이브 베이즈 분류 기반의 핫 데이터 구분 기법

이혜림, 윤이빈, 박동철
숙명여자대학교 소프트웨어학부
{gpfla7946, yibin99, dpark}@sookmyung.ac.kr

Hot Data Identification based on Naïve Bayes Classifier

Hyerim Lee, Yibin Yun, Dongchul Park
Department of Software, Sookmyung Women's University

요 약

최근 낸드 플래시 메모리 기반의 Solid State Drive(SSD)가 기존 Hard Disk Drive(HDD)를 대신하여 개인용과 산업용으로도 널리 쓰이고 있다. 핫 데이터 구분 기법은 이러한 SSD의 성능과 수명에 중요한 역할을 하는 Garbage Collection(GC)과 Wear Leveling(WL) 기술의 기반이 된다. 본 논문에서는 핫 데이터를 예측하기 위한 나이브 베이즈 분류 기반의 새로운 핫 데이터 구분 기법을 제안한다. 제안 기법은 워크로드 액세스 패턴의 학습 단계인 초기 단계와 실제 운영 단계를 통해 다시 액세스 될 확률이 높은 데이터를 그렇지 않은 데이터와 효과적으로 구분한다. 다양한 실제 trace 기반 실험을 통해 본 제안 기법이 기존 대표적인 기법보다 평균 19.3% 높은 성능을 확인했다.

1. 서론

최근 각광받는 Solid State Drive(SSD)를 비롯한 낸드 플래시 메모리 기반 저장 장치는 근본적으로 덮어쓰기(in-place update)가 지원되지 않으며 데이터를 지울 수 있는 횟수의 한계가 존재한다. Flash Translation Layer(FTL)는 논리적-물리적 주소변환을 통해 덮어쓰기가 가능한 것처럼 낸드 플래시 기반 저장 장치를 emulate 해주는 매우 중요한 소프트웨어 계층이다. 또한 FTL은 Garbage Collection(GC)과 Wear Leveling(WL) 메커니즘을 통해 주기적으로 유효하지 않은 데이터들을 효율적으로 삭제함으로써 낸드 플래시 메모리의 성능과 수명에 결정적인 역할을 담당한다[1]. 이 GC와 WL 알고리즘은 기본적으로 핫 데이터 구분 기법에 기초한다. 핫 데이터 구분 기법은 자주 액세스되거나 변경되는 데이터(핫 데이터)를 그렇지 않은 데이터(콜드 데이터)들과 구분하는 기법이다[3]. 즉, 가까운 미래에 핫 할 데이터를 효과적으로 구분하고 관리함으로써 낸드 플래시 메모리의 수명을 연장하고, 성능을 향상시킬 수 있다. 따라서 기존 다양한 핫 데이터 구분 기법들이 연구되어 왔으며[2][3], 최근에는 딥러닝 기법들을 적용한 연구가 이루어지고 있는 추세이다[4][5]. 그러나 딥러닝 기반의 기법들은 각 trace마다 모델을 별도로 만들어 트레이닝 시켜야 하므로 워크로드의 특성이 변화하면 모델의 성능이 저하되는 한계점이 있다.

본 논문에서는 여러 trace에 범용적으로 적용 가능

하며, 변화하는 trace의 특성에 동적으로 적응할 수 있는 새로운 핫 데이터 구분 기법을 제안한다. 제안 기법은 나이브 베이즈의 개념을 이용함으로써 적은 연산량과 빠른 예측이 가능하므로 핫 데이터 구분의 온라인 처리가 가능하다. 성능을 검증하기 위하여 다양한 실제 trace를 기반으로 실험을 진행한 결과 기존 구분 기법 대비 최고 33%의 성능 향상을 확인했다.

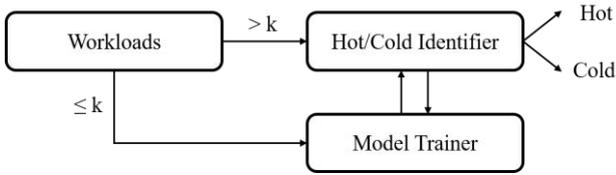
2. 관련 연구

기존 핫 데이터 구분 기법들로는 다중 해시 함수 기법(Multiple Hash Function)[2]과 다중 블룸 필터 기법(Multiple Bloom Filter)[3]이 가장 대표적이다. MHF는 여러 개의 해시 함수를 사용해 빈도수(frequency)를 효율적으로 잡아내는 핫 데이터 구분 기법이다. 그러나 최신성(recency)을 고려하지 않아 상대적으로 분류의 정확도가 떨어진다는 한계가 있다. 반면 MBF는 여러 개의 블룸 필터에 서로 다른 가중치를 부여함으로써 빈도수 뿐만 아니라 최신성까지 효과적으로 고려하여 핫 데이터를 분류한다. 그러나 MHF와 MBF는 미래의 핫 데이터를 예측한 것이 아닌, 현재의 핫 데이터를 분류한 기법이다. 최근에는 딥러닝 기법인 Long Short-Term Memory(LSTM)를 이용한 두 가지 기법들이 제안되었다. LSTM+KM[4]는 LSTM과 K-Means 클러스터링 기법을 이용하여 Logical Block Address(LBA)들의 미래 온도를 예측함으로써, SSD의 GC 성능향상을 목

적으로 제안되었다. Pseudo-OPT[5]는 LSTM 을 이용하여 같은 LBA 가 다시 액세스 되는 시기를 예측하여 Belady 의 optimal algorithm[5] 에 적용함으로써, 캐시 성능 향상에 목적이 있다. 그러나 이 두 기법은 trace 마다 사전에 모델을 트레이닝 해야 한다는 단점이 존재한다.

3. 제안 방법

본 논문에서 제안한 기법의 전체적인 프레임워크는 (그림 1)과 같다.



(그림 1) 전체적인 프레임워크

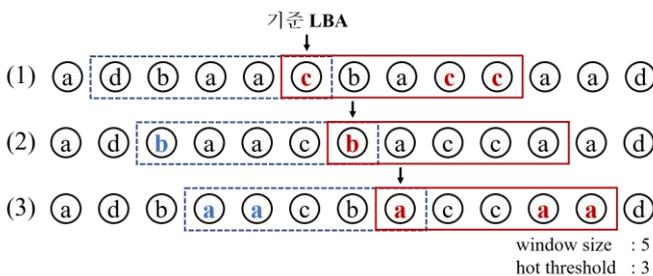
워크로드에서 초기 k 개의 request 는 핫 데이터를 분류하지 않고, 나이브 베이즈 분류기 학습에 사용된다. 이후 request 는 각 속성값들을 나이브 베이즈 분류기에 대입하여 핫 데이터가 될 확률과 콜드 데이터가 될 확률을 계산하고 더 높은 확률값을 가진 클래스로 분류한다. 그리고 구분한 결과를 나이브 베이즈 분류기에 피드백 한다.

3.1 속성 변수

제안한 기법의 나이브 베이즈 분류기는 세 가지 속성 변수를 사용한다: 빈도수(Freq), reuse distance(RD), penultimate reuse distance(Pen.RD)[5]. 이 속성 변수들의 값은 정해진 크기의 윈도우 내에서 계산된다. 빈도수는 현재 LBA 가 액세스 된 횟수이다. Reuse distance 는 동일 LBA 액세스 사이의 다른 LBA 액세스들의 개수이다. Penultimate reuse distance 는 동일 LBA 의 바로 직전 액세스가 아닌 두 번째 전 액세스 사이의 reuse distance 이다.

3.2 메인 디자인

- 초기 단계



(그림 2) 초기 단계에서의 슬라이딩 윈도우 예제

<표 1> 속성 변수들과 label 의 값 예제

	Freq	RD	Pen.RD	Label
(1) c	1	* N/A	* N/A	Hot
(2) b	2	3	* N/A	Cold
(3) a	3	2	3	Hot

* window 범위 내 없음

제안 기법에서는 초기 k 개의 request 로 나이브 베이즈 분류기를 학습시킨다. 보다 정확한 학습을 위해 (그림 2)와 같은 과거와 미래 윈도우가 존재하고, 두 윈도우에 중복된 유일한 LBA 를 기준 LBA 라고 한다. 과거 윈도우를 이용하여 기준 LBA 의 속성 변수 값들을 구한다. <표 1>은 (그림 2)의 속성 변수 값들을 구한 예제이다. 미래 윈도우에서는 기준 LBA 와 가까운 값일 수록 더 큰 가중치를 부여하여 가중 빈도수를 구한다. 가중 빈도수가 핫 데이터로 분류되는 기준치보다 크면 핫으로, 그렇지 않을 경우 콜드로 분류한다. 이렇게 구한 속성 변수 값들과 분류(label)를 기준으로 나이브 베이즈 분류기를 점진적으로 학습시킨다.

- 운영 단계

초기 단계 이후 모든 request 에 대해 나이브 베이즈 분류기를 이용하여 핫 데이터를 구분하는 운영단계가 진행된다. 단, 운영 단계에서는 미래 윈도우는 더 이상 유지하지 않는다.

$$P(T|f_1, f_2, f_3) = \frac{P(T) \cdot P(f_1, f_2, f_3|T)}{P(f_1, f_2, f_3)}$$

$$\approx \frac{P(T) \cdot P(f_1|T) \cdot P(f_2|T) \cdot P(f_3|T)}{P(f_1, f_2, f_3)} \quad (1)$$

본 기법에서 나이브 베이즈 분류기는 세 가지 속성 변수들 (f_1, f_2, f_3)의 독립을 가정한 식(1)을 사용한다. 그리고 식(2)를 이용하여 확률이 가장 큰 클래스(핫, 콜드)로 분류하고, 분류 결과를 다시 나이브 베이즈 분류기에 반영한다.

$$C_{NB} = \underset{j \in \{1,2\}}{\operatorname{argmax}} [P(T_j) \prod_{i=1}^3 P(f_i|T_j) \times \alpha], \quad (2)$$

$$C_{NB} = \{hot, cold\}$$

$$\alpha = \frac{n(T_j) + n(T_j \cup T_{3-j})}{n(T_{3-j})} \quad (3)$$

여기서 식(2)의 α 는 제안 기법의 성능을 최적화 하기 위한 제어 변수이다 (식(3) 참조). α 는 워크로드를 입력 받는 동안 변화하는 동적 특성이 나이브 베이즈 분류기에 보다 적절하게 반영되도록 가중치를 조정한다.

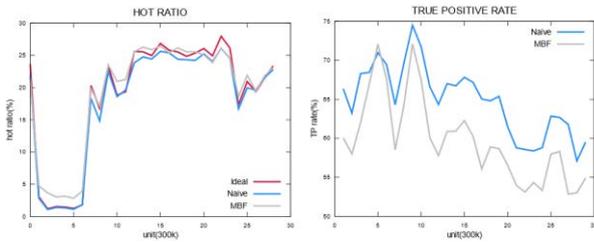
4. 실험 결과

본 절에서는 앞서 제안한 기법의 성능을 검증하기 위한 다양한 실험을 진행한다. 미래를 알고 있다고 가정된 정답값(label)으로 baseline 모델을 만들고 각 기법들의 예측 결과값을 검증하기 위해 사용한다. 기존에 제안된 대표적인 핫 데이터 구분 기법인 다중 블룸 필터 기반의 기법(MBF)과, 본 논문에서 제안한 나이브 베이즈 기반의 기법(Naïve)을 baseline 과 동시에 비교한다. 실험을 위해 Financial 과 MSR trace[3]를 이용했다.

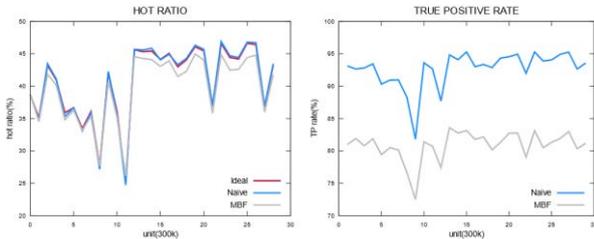
<표 2> 실험 변수와 값

	MBF	Naïve Bayes	Baseline
블룸 필터 크기	4096	N/A	N/A
블룸 필터 개수	4	N/A	N/A
초기화 시간	1024	N/A	N/A
핫 데이터 값	4	4	4
윈도우 크기	N/A	4096	4096
k	N/A	1000000	N/A
가중치 차이	0.5	N/A	0.25

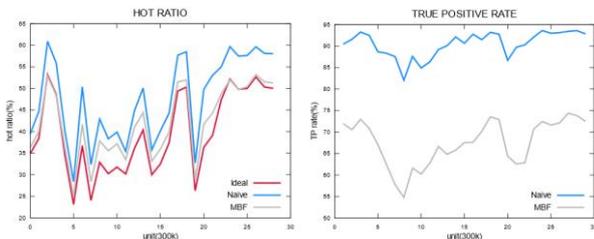
MBF 와의 공정한 비교를 위해 <표 2>와 같이 실험 변수의 값들을 설정했다. 성능 측정 기준으로 핫 데이터 비율 뿐만 아니라 True Positive Rate(TPR)을 채택했다. 이는 각 기법들의 동일한 핫 데이터 비율이 반드시 동일한 분류 성능을 의미하지 않기 때문이다. 따라서 본 실험에서는 각 기법들의 분류값이 baseline 모델의 분류값과 얼마나 동일한지 척도를 보여주는 TPR 을 함께 측정했으며, 각 기법들의 TPR 이 성능의 매우 중요한 척도가 된다.



(그림 3) financial1 실험 결과



(그림 4) MSRrsrch 실험 결과



(그림 5) MSRRusr 실험 결과

(그림 3)과 (그림 4)에서 보듯, 핫 데이터의 비율은 두 기법 모두 baseline 과 평균적으로 4% 미만의 차이로 매우 유사하다. 그러나 TPR 은 MBF 보다 Naïve 가 각 trace 에서 평균적으로 각각 10%와 15% 높았다.

(그림 5)에서는, Naïve 의 핫 데이터의 비율이 MBF 보다 평균적으로 13% 높았으나 TPR 은 Naïve 가 MBF 보다 평균적으로 33% 높았다.

실험에서 보듯 본 제안 기법(Naïve)이 기존 MBF 기법과 핫 데이터 분류 비율은 비슷하나 더 높은 TPR 값을 보여주므로 실제 예측에 대한 정확도는 본 제안 기법이 더 높음을 확인했다.

5. 결론

본 논문에서는 나이브 베이즈 분류기를 이용한 새로운 핫 데이터 분류 기법을 제안하였다. 제안 기법은 초기 단계와 운영 단계를 거치며 핫 데이터를 분류하며, 분류된 결과를 나이브 베이즈 분류기에 피드백해서 점진적 학습을 가능하도록 한다. 실험을 통해 제안 기법은 baseline 의 분류 결과와 최고 93% 일치하는 정확성을 확인했다.

Acknowledgment

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 소프트웨어중심대학지원사업(No. 2022-0-01087) 과 한국연구재단(No. 2020R1F1A1048485)의 지원을 받아 수행되었음.

참고문헌

- [1] Z. Fan and D. Park. “Extending SSD Lifespan with Cooperative Non-Volatile Memory-based Write Buffers.” Journal of Computer Science and Technology (JCST), Vol.34, No.1, pp.113–132, January 18, 2019.
- [2] J.-W. Hsieh, T.-W. Kuo, and L.-P. Chang, “Efficient Identification of Hot Data for Flash Memory Storage Systems,” ACM Transactions on Storage, vol. 2, no. 1, pp. 22–40, 2006.
- [3] D. Park and D. H. C. Du, “Hot data identification for flash-based storage systems using multiple Bloom filters,” in Proc. IEEE 27th Symp. Mass Storage Syst. Technol. (MSST), Denver, CO, USA, 2011, pp. 1–11.
- [4] P. Yang, N. Xue, Y. Zhang, Y. Zhou, L. Sun, W. Chen, Z. Chen, W. Xia, J. Li, and K. Kwon, “Reducing garbage collection overhead in SSD based on workload prediction,” in Proc. 11th USENIX Workshop Hot Topics Storage File Syst., Renton, WA, USA, 2020, pp. 1–8
- [5] P. Li and Y. Gu, “Learning forward reuse distance,” arXiv:2007.15859., 2020