

저자 (Authors)	유진선, 김윤아, 김지홍 JinSun Yoo, Yoona Kim, Jihong Kim
출처 (Source)	한국정보과학회 학술발표논문집 , 2019.12, 1135-1137(3 pages)
발행처 (Publisher)	한국정보과학회 The Korean Institute of Information Scientists and Engineers
URL	http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09301853
APA Style	유진선, 김윤아, 김지홍 (2019). 스마트 SSD를 위한 성능평가 모델 및 활용. 한국정보과학회 학술발표논문집, 1135-1137
이용정보 (Accessed)	서울대학교 147.46.242.*** 2020/08/06 17:28 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

스마트 SSD를 위한 성능평가 모델 및 활용

유진선^o, 김윤아, 김지홍

서울대학교 컴퓨터공학부

{jsyoo, yoonakim, jihong}@davinci.snu.ac.kr

An Assessment Model and its Usage for SmartSSD

JinSun Yoo^o, Yoona Kim, Jihong Kim

Department of Computer Science and Engineering, Seoul National University

요 약

빅데이터 시대에서 대량의 데이터가 생산되면서 이를 분석하고 활용하는 것이 중요해졌다. 데이터를 전송하는 병목이 작용하면서 성능 향상에 제한을 보이고 있다. 이러한 문제점을 극복하기 위해 저장장치 근처에서 계산을 미리 수행해 전송되는 데이터를 최소화하는 데이터 근접 연산 (Near Data Processing, NDP)에 관한 연구가 진행되었으며 일례로 삼성전자에서 상용화 중인 SmartSSD를 들 수 있다. 그러나 임의의 상황에서 SmartSSD를 비롯한 NDP를 사용하는 것이 유의미한 성능 향상을 가져오는지 평가하는 모델이 없다. 이에 본 논문에서는 SmartSSD의 성능을 이론적으로 평가 및 분석하는 모델을 제시하고 이를 검증한다. 모든 경우에서 SmartSSD로 인한 성능 향상이 이루어지지 않는다는 것을 확인했다.

1. 서 론

빅데이터 시대가 도래하며 대량의 데이터를 분석해 의미 있는 통찰을 얻는 것이 중요해졌다. 빅데이터 분석은 현재 k소셜 네트워크 그래프, 패턴 예측, 고객 데이터 분석, 추천 알고리즘 등 다양한 분야에 사용되고 있다.[1] 한 통계에 의하면 페이스북은 매일 600TB의 데이터를 생성하며 300PB에 달하는 데이터로부터 지속적으로 의미 있는 결과를 얻어내는 데이터 분석을 진행하고 있다. [2]

이러한 데이터 분석을 실행하기 위해 계산 성능, 빠른 데이터 전송 속도, 그리고 빠른 저장장치가 필요하다. 이러한 요소 중 하나라도 제한된다면 데이터 분석 과정에서의 성능의 한계점을 마주할 수 있다. CPU가 마주하는 계산 성능의 한계는 GPU나 FPGA와 같은 하드웨어 가속기로 극복한 반면 데이터 전송 속도는 여전히 병목으로 작용하고 있다. 데이터 분석을 위해 저장장치에서 가속기로 데이터를 전송하는 과정에 있어 사용되는 PCIe 인터페이스의 대역폭이 전체 과정의 대역폭을 제한한다. 이로 인해 하드웨어 가속기의 연산 성능 향상이 실 성능 향상으로 이어지지 못하고 있다.

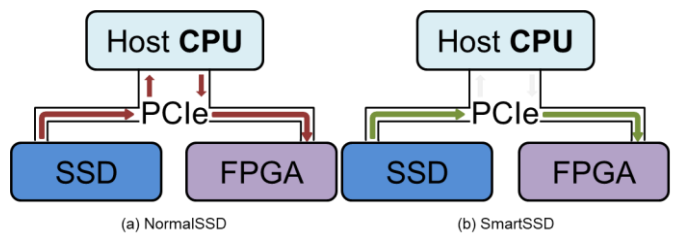
이러한 문제를 극복하기 위해 선행연구는 계산 연산을 저장장치에서 수행하여 데이터 전송을 최소화하는 데이터 근접 연산 (Near Data Processing, NDP)을 제시하였다. SSD의 메인 프로세서에서 데이터를 미리 선별해 PCIe를 통해 전송되는 데이터의

양을 줄이는 연구가 이루어졌으며[3-4] 최근에는 계산 성능의 한계가 명확한 프로세서 대신 FPGA를 활용하는 연구도 진행되었다[5]. 이런 개념을 실현화 한 것으로 삼성전자에서 제시한 SmartSSD가 있다.

선행 연구는 데이터 전송 과정에서의 병목을 완화하는 방법을 제시하고 실험을 통해 그 효과를 입증했으나, 일반적인 상황에서 SmartSSD를 사용해 실익을 거두는지를 평가할 수 있는 모델을 제시하지 못했다는 한계가 있다. 이에 본 논문은 Near Data Processing을 사용하는 것에 대해 판단할 수 있는 평가 모델을 제시하고 이를 검증한다. 검증은 삼성전자에서 상용화한 SmartSSD를 사용한다[6].

본 논문의 구성은 다음과 같다. 2장에서는 SmartSSD를 사용함으로써 데이터 전송 경로가 어떻게 변하는지를 설명한다. 3장에서는 SmartSSD를 사용하는 효과를 평가할 수 있는 모델을 제시하며 4장에서는 다양한 응용을 통해 이를 검증한다.

2. SmartSSD



[그림 1] 일반 상황과 NDP 상황의 데이터 경로 비교

삼성전자에서 제시한 SmartSSD는 SSD와 FPGA 사이에 PCIe peer-to-peer를 통해 데이터를 직접 주고받을 수 있게 했다. FPGA를 사용함으로써 활용할

• 이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다. 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2015M3C4A7065645). (교신저자: 김지홍)

수 있는 데이터 분석의 폭이 상대적으로 넓어졌으며, 특히필요에 따라 FPGA에 프로그램 된 내용을 쉽게 바꿀 수 있다는 장점이 있다.

그림 1은 SmartSSD로 인한 데이터 전송 경로의 변화를 나타내고 있다. 기존에는 SSD에 저장된 데이터를 FPGA에서 처리하기 위해 SSD에서 CPU 메모리로 데이터를 읽어온 후 그 데이터를 다시 FPGA로 전송해야 한다. 그러나 SmartSSD는 SSD와 FPGA 사이에 데이터 전송 경로를 형성함으로써 CPU 메모리의 개입 없이 PCIe 통신 한 번으로 데이터를 바로 전송할 수 있게 되었다. 이를 통해 CPU의 자원이 불필요하게 낭비되는 것을 방지하면서 PCIe 통신의 횟수를 줄임으로써 데이터 전송 병목의 영향을 축소할 수 있게 되었다.

SmartSSD의 작동 방식은 다음과 같다. 프로그래밍의 전체적인 틀은 OpenCL framework 상에서 이루어진다. CPU가 FPGA 메모리에 OpenCL Buffer를 생성한 다음 그 주소를 읽기 명령과 함께 SSD로 전달하면 SSD는 읽어낸 데이터를 CPU의 메모리로 전송하듯 FPGA의 DRAM으로 전송한다. CPU는 SSD에 읽기 명령을 내리고 FPGA의 각종 설정을 하며, 데이터 전송이나 분석 과정을 직접 수행하지는 않는다.

3. 평가 모델

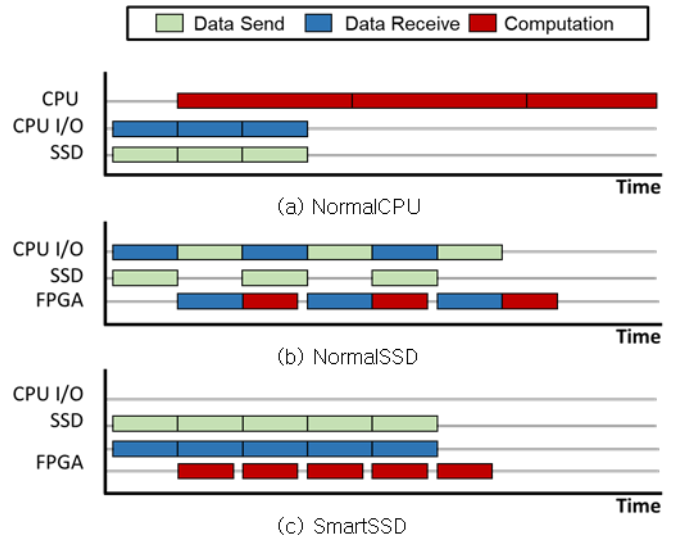
3.1 데이터 전송 경로 별 타임라인 분석

그림 2는 빅데이터 분석 과정에서 사용할 수 있는 세 가지 데이터 전송 경로 별 타임라인을 나타낸다. 각 경로 별 분석 수행은 다음과 같이 이루어진다.

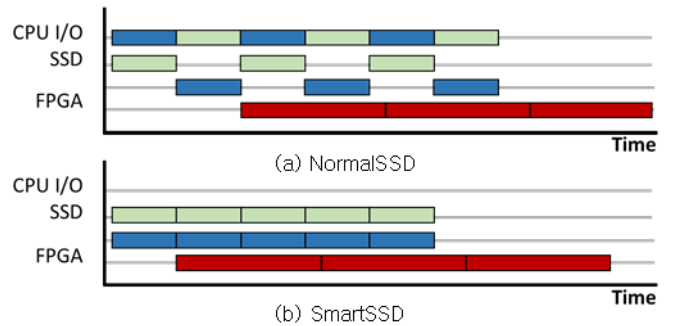
- NormalCPU: 데이터를 SSD에서 CPU로 옮긴 다음 CPU에서 연산을 수행한다.
- NormalFPGA: 데이터를 SSD에서 CPU로 먼저 옮긴 다음, CPU에서 다시 가속기로 전송한다. 가속기에서 가속화된 연산을 수행한다.
- SmartSSD: 데이터를 SSD에서 바로 가속기로 전송한 다음 가속기에서 연산을 수행한다. SmartSSD가 이 경우에 해당한다.

위의 세 경로에서 나타나는 block의 시간은 데이터를 PCIe를 통해 전송하는 시간 시간 $t_{I/O}$, CPU에서 연산을 수행하는 시간 t_{CPU} , FPGA에서 연산을 수행하는 시간 t_{FPGA} 로 설명할 수 있다. NormalCPU의 경우, 데이터 연산 한번 수행 시 SSD에서 CPU로 데이터를 전송하는데 $t_{I/O}$, CPU에서 연산을 수행하는 데 t_{CPU} 가 걸려 총 $t_{I/O}+t_{CPU}$ 가 걸린다. 다른 경로에서의 수행 시간도 다음과 같이 정리할 수 있다.

$$\begin{aligned} t_{NormalCPU} &= t_{I/O} + t_{CPU} \\ t_{NormalSSD} &= 2t_{I/O} + t_{FPGA} \\ t_{SmartSSD} &= t_{I/O} + t_{CPU} \end{aligned}$$



[그림 2] 데이터 전송 경로 별 타임라인



[그림 3] SmartSSD의 효과가 없는 상황의 타임라인

3.2 이론적 분석을 통한 평가 모델 제시

그림 2에서 나타난 데이터 경로 별 작업에 소요되는 수행 시간을 분석한다. 그림에서와 같은 pipelining 상황에서는 각각의 경로에서 하나의 일의 단위를 수행하는 데 소요되는 시간은 아래와 같다.

$$\begin{aligned} t_{NormalCPU} &= \max(t_{I/O}, t_{CPU}) \\ t_{NormalSSD} &= \max(2t_{I/O}, t_{FPGA}) \\ t_{SmartSSD} &= \max(t_{I/O}, t_{FPGA}) \end{aligned}$$

FPGA는 CPU의 연산을 가속하므로 $t_{CPU} > t_{FPGA}$ 임을 가정할 수 있다. 따라서, NormalSSD보다 SmartSSD가 우위에 있는지를 확인하려면 $t_{NormalSSD} > t_{SmartSSD}$ 조건을 만족하는지 확인하면 된다. 이 조건을 풀어 쓰면 아래와 같다.

$$\begin{aligned} t_{NormalSSD} - t_{SmartSSD} &= \begin{cases} t_{FPGA} - t_{FPGA} = 0 & (t_{FPGA} > 2t_{I/O}) \\ 2t_{I/O} - t_{FPGA} & (2t_{I/O} > t_{FPGA}) \\ 2t_{I/O} - t_{I/O} = t_{I/O} & (t_{I/O} > t_{FPGA}) \end{cases} \end{aligned}$$

결론적으로, SmartSSD를 운용하여 효과를 보고자 하던 $2t_{I/O} > t_{FPGA}$ 라는 조건을 만족해야 한다. 그림 3은 해당 조건을 만족하지 못할 경우 SmartSSD를 사용해도 큰 효과가 나타나지 않는 타임라인을 보이고 있다. SmartSSD를 사용한다고 무조건 유의미한 성능을 보장하지 않을 수 있다는 것을 확인했다. 따라서 FPGA의 계산 능력이 CPU 대비 높다고 하더라도 이를 전부 활용하지 않는 비교적 간단한 분석이 SmartSSD 운용에 적당하다고 평가할 수 있다.

4. 실험

4.1 실험 개요

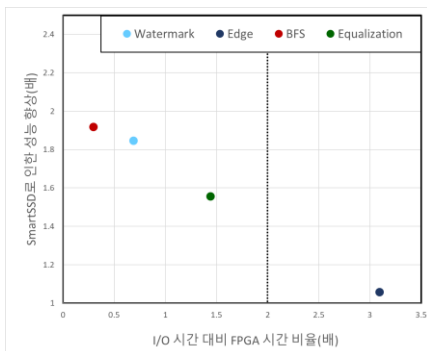
3장에서 제시한 모델을 검증하기 위해 복수의 응용에 대하여 분석을 실시하였다. 먼저, 각 응용 별로 $t_{I/O}$ 와 t_{FPGA} 를 측정하고 다음 그 비율 $r_t = t_{FPGA} / t_{I/O}$ 를 측정한다. 만일 $r_t > 2$ 라면 3장에서 제시한 모델의 조건을 만족한다고 볼 수 있다. 이후 NormalSSD와 SmartSSD 각각에 소요되는 소요시간을 측정하여 시간단축 비율을 구한다. 이를 통해 응용별로 SmartSSD를 사용함으로써 NormalSSD 대비 얻는 성능 향상 효과를 측정한다. 이를 비교해 모델과 실험 결과 사이의 일관성을 확인한다.

실험에 사용하는 응용은 Xilinx에서 제공하는 프로그램과 직접 구현한 프로그램을 함께 사용하며, 해당 프로그램의 목록은 아래 표 1과 같다.[7] SmartSSD에 사용된 SSD는 삼성 NVMe SSD SM981, FPGA는 Xilinx사의 ZynqPlus 19를 사용했으며, 데이터 전송 인터페이스는 PCIe Gen3.0을 사용했다.

FPGA 분석	설명
bfs	그래프에 대한 BFS 탐색 수행
watermark	단순함으로 이미지 변형
edge	소벨 필터를 이용한 edge detection
equalizer	Histogram Equalizer 수행

[표 1] 하드웨어 응용에 따른 FPGA 자원 사용

4.2 실험 결과



[그림 4] 응용 별 성능 평가 결과

그림 4는 각 응용 별로 r_t 와 SmartSSD를 사용할 때의 성능 향상 비율을 나타낸 것이다. 실험 결과 $2t_{I/O} > t_{FPGA}$ 라는 조건을 만족하는지에 따라 성능 향상

비율이 크게 차이 나는 것을 확인할 수 있었다. 점선 왼쪽에 있는 응용들은 3장에서 제시한 조건을 만족하는 응용 들로서, 최소 1.5배 이상 성능 향상 효과를 보였다. 점선 우측에 있는 응용은 해당 조건을 만족하지 못한 응용으로서, 성능 향상이 거의 없음을 볼 수 있었다.

또한, 응용에서 I/O가 차지하는 비중과 SmartSSD를 통한 성능 향상 사이의 경향성도 확인할 수 있었다. r_t 가 작을수록, 응용에서 I/O가 차지하는 비중이 클수록 SmartSSD를 사용한 효과가 크게 나타났다.

5. 결론 및 향후 연구

데이터 분석에 있어 데이터를 전송하는 과정이 인터페이스의 제한으로 인해 병목으로 작용해왔다. 이를 완화하기 위해 저장장치 근처에서 계산을 수행하는 연구가 이루어져왔으나, 일반적인 상황에서 성능을 평가하는 모델이 없다는 문제가 있었다. 본 논문에서는 SmartSSD를 사용함으로써 얻는 효과를 평가하기 위한 모델을 제시하고 이를 검증하였다. SmartSSD가 효과를 거두기 위해서는 특정 조건을 만족해야 하며, 이를 만족하지 못하는 경우 SmartSSD로 인한 성능향상폭이 크지 않다는 것을 볼 수 있었다. 또한, I/O가 차지하는 비중이 클수록 SmartSSD를 이용할 때 성능 향상폭이 커지는 것도 확인할 수 있었다. 향후 연구로는 CPU자원에 유휴가 발생하는 것을 이용하여 FPGA 연산의 일부를 CPU와 함께 분담하는 이기종 데이터 분석 환경에 대한 연구를 진행할 수 있을 것이다.

참고 문헌

[1] G. Malewicz et al., "Pregel: A system for large-scale graph processing", Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 135-146, 2010

[2] Cassidy R. Sugimoto et al., Big Data Is Not a Monolith. The MIT Press, 2016

[3] B. Gu, et al., "Biscuit: a framework for near-data processing of big data workloads", Proceedings of the 43rd International Symposium on Computer Architecture, pp.153-165, 2016

[4] I. Jo et al., "YourSQL: a High-Performance Database System Leveraging In-Storage Computing", Proceedings of the VLDB Endowment, pp.924-935, August 2016

[5] R. Zhenyuan et al., "Insider: designing in-storage computing system for emerging high performance device", Proceedings of the 2019 USENIX Annual Technical Conference, pp.379-394, 2019

[6] Samsung SmartSSD
<https://samsungatfirst.com/smartssd/>

[7] Xilinx SDAccel Examples
https://github.com/Xilinx/SDAccel_Examples