시뮬레이터 기반 iSSD에서 데이터 마이닝 알고리즘 성능 평가

정문준*, 조용연**, 김상욱**†, 오현옥○
*한양대학교 컴퓨터공학부
**한양대학교 컴퓨터·소프트웨어학과
e-mail: *chung1246@hanmail.net
**{jyy0430,wook}@hanyang.ac.kr, ○hoh@hanyang.ac.kr

Data Mining in Intelligent SSD: Simulator Based Evaluation

Moonjun Chung*, Yong-Yoen Jo**, Sang-Wook Kim**†, Hyunok Oh○
*Division of Computer Science and Engineering, Hanyang University
**Dept. of Computer and Software, Hanyang University
○Dept. of Information Systems, Hanyang University, Korea

요 약

Intelligent SSD (iSSD)는 SSD 내부에 프로세서들을 장착하여 데이터를 처리할 수 있도록 한 저장 장치이다. 본 논문은 iSSD 시뮬레이터를 이용하여 다양한 데이터 마이닝 알고리즘을 수행함으로써 iSSD의 가능성 및 효용성을 확인해 보고자 한다.

1. 서론

Solid Storage Device (SSD)는 높은 I/O 대역폭과 낮은 access latency를 가짐으로써 Hard Disk를 대체하는 저장 장치로 각광받고 있다 [1]. 그러나 SSD를 사용함에도 불구하고 기존 처리방식인 In-Host Processing (IHP)의 한계로 인한 문제점이 발생한다. IHP는 데이터를 저장 장치에서 호스트의 메모리 (DRAM) 로 전송한 후, 호스트의 프로세서를 사용하여 데이터를 처리하는 방식이다 [2].

특히 I/O를 자주 요청하는 알고리즘의 경우, IHP 방식은 호스트 인터페이스의 물리적 속도 제한으로 인해 호스트 인터페이스의 병목현상이 발생할 수 있다. 데이터 마이닝과 같은 data-intensive 알고리즘에서는 이러한 병목현상으로 인한 심각한 성능 저하가 발생할 수 있다 [3].

이러한 병목현상을 해결하기 위한 방법으로 In-Storage Processing (ISP)가 제안되었다. ISP 방식은 저장 장치 내부에 프로세서들을 장착하여 이들을 이용해 데이터를 처리하는 방식이다. 최근에는 SSD에서 ISP를 수행하기 위한 intelligent SSD (iSSD)가 제안되었다 [4].

iSSD는 SSD 내부에 프로세서들을 장착하여 데이터를 처리할 수 있도록 한 저장 장치이다. iSSD는 각 채널마다 데이터 저장 장소인 셀, 셀을 관리하기 위한 flash memory controller (FMC), SRAM이 존재한다. 이 채널들은 flash translation layer (FTL) 펌웨어에 의해 제어되며, iSSD에는 FTL을 수행하기 위한 SSD core와 DRAM, SRAM이 존재한다 [5].

본 논문에서는 iSSD 시뮬레이터를 통해 iSSD의 가능성 및 성능을 검증하고자 한다. 따라서 다양한 가상의 컴퓨팅 환경을 구축할 수 있는 Gem 5 시뮬레이터 [6]을 이용함으로써 iSSD 시뮬레이터를 구현한다. 또한 이를 위한 다양한 데이터 마이닝 알고리즘을 구현하고 성능을 측정한다.

본 논문에서는 Gem5 시뮬레이터를 이용하여 iSSD 시뮬레이터를 구현하였다. 또한 iSSD 시뮬레이터를 위해 다양한 데이터 마이닝 알고리즘을 구현하고 이에 대한 성능을 측정하였다. 실험 결과, core의 수가 증가하거나 성능이 향상되었을 때, 알고리즘의 수행시간이 단축되었다.

2. iSSD 시뮬레이터 구현

본 논문에서는 Gem5 시뮬레이터를 사용하여 iSSD 환경을 구현하였다 [6]. Gem5 시뮬레이터는 가상의 컴퓨팅 환경을 구축하여 그 위에서 프로그램을 처리할 수 있는 시뮬레이터이다. Gem5 시뮬레이터에서는 프로세서, 메모리, 캐시의 성능을 조절할 수 있다. 따라서 Gem5의 프로세서, 메모리, 캐시를 iSSD의 SSD core, DRAM와 SRAM으로 간주하여 컴퓨팅 환경을 구현할 수 있다. 시뮬레이터의 정확도를 높이기 위해 프로세서 모델은 AtomicSimple CPU, 그리고 시스템 모드는 System-call Emulation mode (SE) 로 설정하였다 [6].

3. 실험

3.1 실험 환경

본 실험을 수행한 호스트 환경은 Ubuntu 14.04.2 LTS 이고, 프로세서로 intel i5 3470, 8GB RAM, 저장 장치로 intel SSD 530을 사용하였다. 실험에는 표 2의 데이터를 사용하였다. 크기가 큰 데이터를 사용하면 시뮬레이션 시간이 오래 걸리기 때문에 시뮬레이션 시간을 줄이기 위하여 작은 크기의 데이터를 사용하였다.

표 2. 실험 데이터

† 교신저자

본 연구는 (1) 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업 (IITP-2015-H8501-15-1013), (2) 정부(미래창조과학부)의 재원으로 한국연구재단의 지원 (NRF-2014R1A2A1A10054151)과 (3) 한양대-삼성전자 반도체 산학협력 연구 과제의 지원으로 수행됨.

	타입	세부 사항
k-means 데이터	합성	객체 수: 10,000 객체의 차원: 3
Apriori 데이터	합성	item list의 수 (#item): 10,000 item 종류의 수: 20
PageRank 데이터 [7]	실세계	객체 수: 7,115 객체간의 관계의 수: 103,689
Decision Tree 데이터 [8]	실세계	데이터 수 : 850 Attribute 수 : 19

3.2 iSSD 시뮬레이터 기반의 데이터 마이닝 알고리즘 성능 평가

본 실험에서는 iSSD 시뮬레이터에서 데이터 마이닝 알고리즘의 성능을 평가하고자 한다. 이를 위해 본 실험에서는 iSSD 시뮬레이터의 인자를 변화 시켜가며 각 알고리즘을 수행하고 수행 시간을 측정하였다. 각 알고리즘의 결과 그래프에서 각 선은 iSSD 시뮬레이터의 프로세서 성능을 나타낸다. X 축은 iSSD 시뮬레이터의 프로세서 수이며, Y축은 수행시간을 나타낸다.

실험 결과 1 - Apriori

그림 1는 iSSD 시뮬레이터에서 Apriori의 성능을 나타낸 것이다. Apriori는 프로세서 수의 증가와 성능의 향상은 알고리즘의 수행시간이 단축되나 프로세서 수에 비례하여 향상되지는 않는다.

Apriori의 함수들이 병렬 처리 되기 때문에 프로세서 수가 많으면 수행 시간이 단축되었다. 그러나 Apriori 내에 중복된 데이터를 제거하기 위해 단일 스레드로 처리해야 되는 과정이 필요하다. 따라서 이 과정에서 병목현상이 발생하여 프로세서 수에 비례하여 향상되지는 않는다.

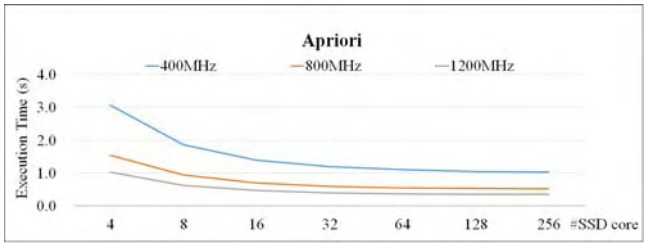

그림 1. iSSD 시뮬레이터에서 Apriori의 성능

실험 결과 2 - k-means

그림 2는 iSSD 시뮬레이터에서 k-means의 성능을 나타낸 것이다. iSSD 환경에서 k-means는 프로세서 수의 증가와 성능의 향상함에 따라 비례하여 수행시간이 단축되었다. 이는 k-means 내의 함수들이 병렬처리가 가능하기 때문이다.

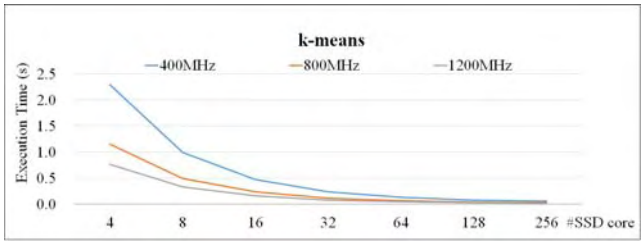

그림 2. iSSD 시뮬레이터에서 k-means의 성능

실험 결과 3 - PageRank

그림 3은 iSSD 시뮬레이터에서 PageRank의 성능을 나타낸 것이다. PageRank는 프로세서 수의 증가와 성능의 향상됨에 따라 그에 비례하여 수행시간이 단축되었다. PageRank 내에는 병렬처리 가능한 함수와 그렇지 않은 함수들이 존재하지만, 병렬처리 가능한 함수의 수행 시간이 전체 수행 시간의 대부분을 차지하기 때문에 프로세서 수에 비례하여 알고리즘의 성능이 향상되었다.

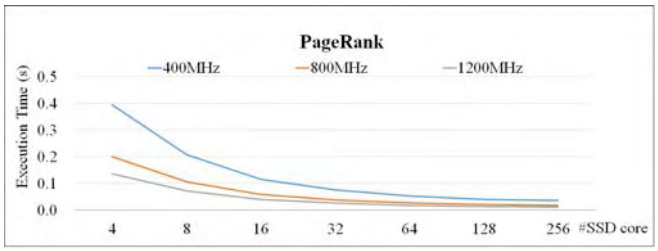

그림 3. iSSD 시뮬레이터에서 PageRank의 성능

실험 결과 4 - Decision tree

그림 4는 iSSD 시뮬레이터에서 Decision Tree의 성능을 나타낸 것이다. Decision Tree는 프로세서 수가 32개일 때까지만 성능이 향상되었다.

이는 데이터의 특성 때문이다. 실험에 사용된 데이터의 attribute의 수는 19개이다. 따라서 프로세서 수가 19개보다 많으면 연산을 수행하지 않는 프로세서가 발생할 수 있다. 이로 인해 프로세서의 수가 32개 이상일 때 성능향상이 없었다. 또한 프로세서의 성능을 증가시켰을 때에는 프로세서가 연산을 더 빨리 수행할 수 있어 알고리즘의 전체적인 성능이 프로세서 성능에 비례하여 향상되었다.

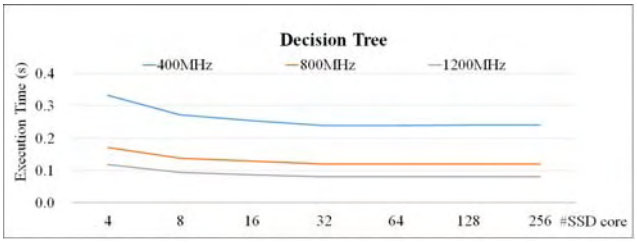

그림 4. iSSD 시뮬레이터에서 Decision Tree

4. 결론

본 논문에서는 iSSD 시뮬레이터를 기반으로 데이터 마이닝 알고리즘을 수행하였다. 실험결과, iSSD 내의 프로세서의 수와 성능이 증가된다면, 병렬처리가 가능한 알고리즘에 대해서 효과적일 것으로 기대된다.

5. 참고 문헌

[1] S. Lee, et al., “A Case for Flash Memory SSD in Enterprise Database Applications,” In ACM SIGMOD, pp. 1075-1086, 2008.
[2] D. Bae et al., “Intelligent SSD: A Turbo for Big Data Mining,” In ACM CIKM, pp. 1553-1556, 2013.
[3] J. Do et al., “Query Processing on Smart SSDs: Opportunities and Challenges,” In ACM SIGMOD, pp.1221-1230, 2013.
[4] S. Kim et al., “Fast, Energy Efficient Scan inside Flash Memory SSDs,” In ADMS, 2011.
[5] Y. Jo et al., "On Running Data-Intensive Algorithms with Intelligent SSD and Host CPU: A Collaborative Approach," In ACM SAC, pp. 2060-2065, 2015.
[6] N. Binkert et al., “The Gem5 Simulator,”ACM SIGARCH Computer Architecture News, pp. 1-7, 2011.
[7] Wiki vote dataset, http://snap.stanford.edu/data/
[8] Complete prepared case study business data sets, http://www.wiley.com/legacy/compbooks/soukup/downloads.html