

Intelligent SSD를 위한 효율적인 외부 정렬 방안

배덕호*, 김진형*, 정교성*, 김상욱*, 박찬익**

*한양대학교 전자컴퓨터통신공학과

**삼성전자

e-mail:dhbae@agape.hanyang.ac.kr

Efficient External Sorting for Intelligent SSD

Duck-Ho Bae*, Jin-Hyung Kim*, Kyosung Jeong*, Sang-Wook Kim*, Chanik Park**

*Dept of Electronics and Computer Engineering, Hanyang University

**Semiconductor Business, Samsung Electronics Co.

요 약

최근, 빅데이터 등 대용량의 데이터가 등장함에 따라 SSD 안에 추가적인 프로세서를 장착하여 데이터 처리 능력을 부여한 Intelligent SSD의 필요성이 대두되고 있다. 본 논문에서는 먼저, Intelligent SSD의 특징을 분석하고, Intelligent SSD가 대용량 처리에 적합함을 보인다. 더 나아가, Intelligent SSD를 위한 외부 정렬 방안을 제안하고, 비용 모델을 수립을 통해 제안한 방안의 우수성을 검증한다.

1. 서론

최근, SSD는 빠른 I/O 속도, 비휘발성, 저전력 소비 등의 장점으로 인해 하드 디스크를 대체하는 저장 장치로 각광받고 있다. 그러나 SSD의 사용은 SSD의 높은 내부 대역폭으로 인해 새로운 병목점 (bottleneck)을 야기하게 된다[1]. SSD는 기계적인 부분이 존재하지 않고, multi-way, multi-channel 기술로 인해 높은 내부 대역폭을 가진다. 반면, 현재 호스트 인터페이스는 상대적으로 낮은 대역폭을 가지고 있다. 이로 인해 SSD의 I/O 속도가 오히려 호스트 인터페이스의 대역폭에 제약받게 된다[2].

본 논문에서는 먼저 SSD의 높은 내부 대역폭을 활용하기 위해 SSD 안에 추가적인 범용 프로세서를 장착하여 데이터 처리 능력을 부여한 Intelligent SSD (ISSD)에 대해 논의한다. 더 나아가, ISSD의 구조적 특징을 분석하고, ISSD에서 데이터 의존적인 프로그램 수행의 적합성을 보인다. 이를 바탕으로, ISSD를 위한 효율적인 외부 정렬 방안을 제안한다. 끝으로, 비용 모델 수립을 통해 제안 방안의 우수성을 보인다.

2. Intelligent SSD

그림 1은 ISSD의 구조를 나타낸다. ISSD는 각 FMC마다 범용 프로세서와 DRAM을 추가함으로써, 내부에서 데이터를 병렬적으로 처리할 수 있는 SSD를 나타낸다[2].

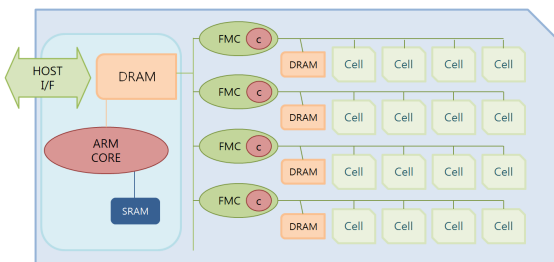


그림 1. ISSD의 구조.

ISSD는 현재 존재하지 않는다. 그러나 ISSD 내에 장착될 범용 프로세서의 성능과 DRAM의 크기는 호스트 프로세서와 비교할 때 다음과 같을 것으로 예상된다.

- 프로세서 성능: 호스트 >> CORE > FMC
- DRAM 크기: 호스트 >> CORE > FMC

ISSD의 특징은 다음과 같다. 첫째, 여러 FMC들을 이용하여 ISSD 내부에서 데이터를 병렬 처리할 수 있게 됨에 따라 SSD의 높은 내부 대역폭을 활용할 수 있게 된다. 이로 인해, SSD에 저장된 데이터를 빠르게 접근할 수 있게 된다. 둘째, 호스트 프로세서로의 데이터 전송량을 줄일 수 있다. ISSD가 데이터 처리 능력을 갖게 됨에 따라 모든 데이터를 전송하지 않고, ISSD에서 미리 처리하여 호스트 프로세서에서 필요한 데이터만을 전송할 수 있게 된다.

이렇듯, ISSD는 빈번한 데이터 접근 작업을 수행하는데 매우 적합하다. 특히, 외부 정렬은 정렬 과정에서 반복적인 데이터의 접근이 빈번히 발생한다. 이러한 작업의 경우, 데이터 병렬 처리의 확장성이 높으므로, ISSD 내에서 외부 정렬을 효율적으로 수행할 수 있을 것으로 예상된다.

3. ISSD를 위한 외부 정렬 방안

본 장에서는 위에서 분석한 ISSD 구조적 특성을 바탕으로 ISSD를 위한 외부 정렬 방안을 제안한다. 외부 정렬이란 메모리에 전부 적재할 수 없는 대용량의 파일을 저장 장치를 이용하여 정렬하는 방안으로, 크게 런 파일 생성 단계와 런 파일 병합 단계로 구성된다[3].

본 논문에서는 런 파일 병합 과정에 대해서만 다루고자 한다. 이는 런 파일 생성 과정의 경우, 메모리 크기가 성능에 절대적인 영향을 미치게 된다. 따라서 런 파일 생성은 메모리 크기가 상대적으로 매우 큰 호스트에서 수행한다고 가정한다.

런 파일 병합은 정렬된 런 파일들을 병합하여 더 큰 런 파일을 생성하고, 이 과정을 반복함으로써 전체 데이터를 정렬하는 과정이다[3]. 해당 과정에서는 디스크 접근 횟수를 줄이기 위해 병합 단계의 최소화에 초점을 맞춘다.

본 논문에서는 효율적인 ISSD 내에서 효율적인 런 파일 병합을 위해 *divide and conquer 전략*[4]을 사용한다. 그림 2는 제안하는 런 파일 방안을 나타낸다. 각 FMC는 자신의 플래시 메모리 셀의 런 파일들을 병합하고, 가장 작은 값들을 순차적으로 CORE로 전송한다. CORE는 각 FMC에서 전송된 결과를 병합하여 병합된 결과를 ISSD에 저장한다.

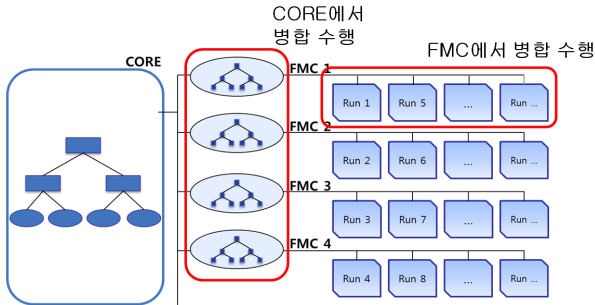


그림 2. 런 파일 병합 방안.

이러한 전략을 통해, 우리는 각 FMC에서 병렬적으로 런 파일 병합을 수행할 수 있게 된다. 또한, 각 FMC의 병합 결과물을 (1) FMC보다 높은 처리 능력과 많은 DRAM 크기를 가지고 있으며, (2) 모든 FMC들과 직접 통신이 가능한 CORE를 이용하여 병합함으로써, 효율적인 전체 병합을 수행할 수 있게 된다.

이 때, 병합의 효율을 위해 FMC와 CORE 모두 병합 트리 인덱스를 사용한다. FMC의 트리의 리프 노드의 수는 해당 채널에 저장된 런 파일의 수와 동일하며, CORE의 리프 노드의 수는 ISSD 내의 채널수와 동일하다.

4. 비용 모델 수립 및 성능 예측

4.1 비용 모델 수립

본 논문에서는 제안하는 외부 정렬 방안의 성능을 예측하기 위해 비용 모델을 수립하였다. 이를 위해, (1) 기존의 호스트 프로세서에서 모두 외부 정렬을 수행하는 모델 (In Host Processing, IHP)과 (2) ISSD 내의 CORE와 FMC를 이용하여 외부 정렬을 수행하는 모델 (In Storage Processing, ISP) 총 2가지 경우에 대한 비용 모델을 수립하였다.

외부 정렬의 수행비용은 크게 (1) 데이터 처리 비용과 (2) 데이터 전송 비용으로 이루어진다. 데이터 처리 비용은 해당 데이터를 처리하는 프로세서의 성능에 직접 영향을 받으며, 데이터 전송 비용은 데이터를 요구하는 장소에 영향을 받는다. 본 논문에서는 이러한 비용의 차이를 비용 모델에 반영한다. 이를 위해, 데이터 처리 비용의 경우, 각 프로세서의 CPU rate에 비례하게 비용을 설정하였으며, 데이터 전송 비용의 경우, SSD 스펙에 근거하여 비용을 설정하였다.

수립한 비용 모델의 정확도 검증하기 위해 IHP 모델에 대한 실제 수행 시간과 비용 모델로 계산된 수행 시간을 비교하였다. 그 결과 수립한 비용 모델은 약 95%의 정확도를 보였다.

4.2 실험 환경 및 결과

본 실험에서는 ISSD를 위한 외부 정렬 방안의 우수성을 검증하기 위해, ISSD 내의 채널의 수를 16개, 32개, 48개로 증가시켜가며, IHP 모델과 ISP 모델의 수행 시간을 비용 모델을 통해 계산하였다. 이 때, 호스트 프로세서, CORE, FMC의 CPU rate는 각각 2.53GHz, 400MHz, 200MHz로 설정하였다.

그림 3은 실험 결과를 나타낸다. 실험 결과, ISP 모델이 IHP 모델에 비해 모든 경우에 있어 성능이 우수함을 알 수 있다. ISP 모델에서 채널수가 32개 이상으로 증가하면 오히려 성능이 나빠진다. 이는 채널수가 증가로 인해 CORE에서의 전체 병합의 작업량이 증가하기 때문이다.

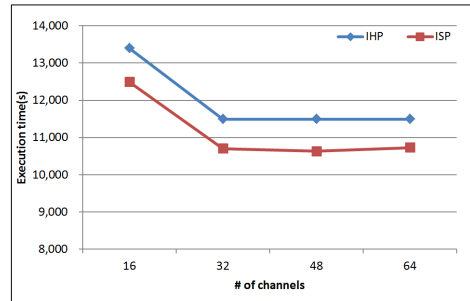


그림 3. IHP와 ISP의 성능 비교.

5. 결론

본 논문의 공헌은 다음과 같다. 첫째, ISSD의 특징을 분석하고, ISSD에서 데이터 의존적인 응용 프로그램 수행의 적합성을 보였다. 둘째, 이를 바탕으로 ISSD를 위한 효율적인 외부 정렬 방안을 제안하였다. 끝으로, 비용 모델 수립을 통해, 제안하는 방안의 우수성과 가능성을 보였다.

감사의 말

“본 연구는 (1) 한양대-삼성전자 반도체 산학협력 연구 과제, (2) 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단 (No. NRF-2012047724), (3) 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업 (NIPA-2013-H0401-13-1001)의 연구비 지원을 받았음”

참고문헌

[1] S. Lee, “A Case for Flash Memory SSD in Enterprise Database Application,” *ACM SIGMOD*, pp. 1075-1086, 2008.
 [2] S. Kim et al., “Fast, Energy Efficient Scan inside Flash Memory SSDs,” *ADMS*, 2011.
 [3] V. Estivill-Castro and D. Wood, “A Survey of Adaptive Sorting Algorithms,” *ACM Computing Surveys*, 1992.
 [4] T. Cormen et al., *Introduction to Algorithms*, The MIT Press, 2000.