# 플래시 SSD의 병렬성을 활용한 외부 합병정렬 개선*

이준희‡ 박상현¶
‡¶연세대학교 컴퓨터과학과
¶교신저자
{joonnc, sanghyun}@cs.yonsei.ac.kr

# Improving External Mergesort by Utilizing Parallelism of FlashSSD

Joonhee Lee‡ Sanghyun Park¶
‡¶Department of computer science, Yonsei University

**요 약**

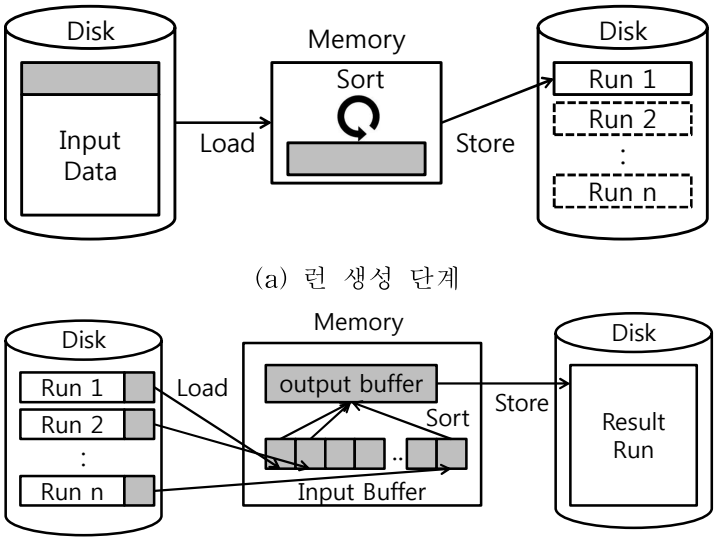
합병정렬은 가장 널리 알려진 외부정렬 알고리즘으로, 정렬하려는 데이터가 사용가능한 메모리보다 더 클 때 사용된다. 과거 합병정렬을 개선하고자 하는 연구가 활발하게 일어났으나 이러한 연구는 합병정렬이 하드디스크에서 수행된다는 전제하에 진행되었다. 이중 가장 일반적인 방법은 이중 버퍼링으로 각 런 당 2개의 독립적인 입력버퍼를 두는 방식을 취한다.

최근 플래시SSD는 차세대 저장매체로 대두되었으며 각종 서버의 저장장치로 채택되기도 하였다. 플래시SSD는 기계적인 움직임이 없어 하드디스크보다 훨씬 빠른 접근시간을 갖으며 훨씬 높은 I/O 대역폭을 발휘할 수 있다. 본 논문은 플래시SSD에서 이중 버퍼링보다 더 높은 성능을 보일 수 있는 합병정렬을 제안한다. 이 논문에서 제안하는 합병정렬은 합병에 필요한 데이터 블록의 순서를 런 생성 단계에서 미리 계산하고, 합병 단계에서 이 순서를 이용해 여러 런으로부터 각각 비동기 I/O를 요청함으로써 플래시SSD의 내부 병렬성을 최대화한다.

## 1. 서 론

합병정렬(mergesort)은 가장 많이 사용되는 외부 정렬 알고리즘 중 하나로, 메모리보다 큰 데이터를 정렬할 때 사용된다[1]. 합병정렬은 (그림 1)과 같이 전체 데이터를 각각의 정렬된 n 개의 런(run)으로 분할하는 런 생성 단계(run generation phase)와, 생성된 런을 하나의 정렬된 런으로 합치는 합병 단계(merge phase)로 구성된다. 합병정렬은 DBMS 등 수많은 응용 프로그램에서 사용되기 때문에 많은 연구가 이루어져 왔다. 합병정렬은 I/O의 성능이 전체 성능에 가장 큰 영향을 미치기 때문에 이러한 연구는 I/O 시간을 줄이는 방향으로 진행되었다. 합병정렬의 I/O 처리(읽기/쓰기)와 CPU 작업(비교 연산)을 동시에 수행하기 위한 방법으로 이중 버퍼링(double buffering)을 사용하는 방법이 제시되었다. 이중 버퍼링은 합병 단계에서 각 런에 2개의 독립적인 입력버퍼를 둔다. 하나의 버퍼안의 데이터가 정렬되는 동안 다른 버퍼에는 해당 런의 다음 블록을 비동기 I/O(Asynchronous I/O)를 통해 메모리로 읽는다. 따라서 전체 입력버퍼의 반은 정렬에, 나머지 반은 읽기에 사용된다.

반면 플래시SSD는 차세대 저장장치로 부각되고 있는 보조기억장치로 여러 개의 플래시 패키지가 (그림 2)와 같이 연결되어 있다. n개의 플래시 메모리 패키지가 하나의 채널에 각각 연결되고 m개의 채널이 SSD의 컨트롤러에 연결된다. 이러한 구조에 기인하여 플래시SSD는 내부 병렬성[2]이라는 특징을 갖는다.




(a) 런 생성 단계

(b) 합병 단계

그림 1. 외부 합병정렬

* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2012R1A2A1A01010775).

각각의 플래시 메모리 패키지가 모두 I/O에 병렬적으로 참여하게 되면 플래시SSD의 대역폭(bandwidth)을 최대로 사용할 수 있다는 것이다. 그러나 이러한 상황은 일반적으로 쉽게 발생하지 않으며 여러 개의 비동기 I/O 요청이 단시간에 플래시SSD에 전달될 때, 그리고 요청된 각각의 I/O 단위가 클 때 발생한다. 이러한 플래시SSD의 내부 병렬성을 최대로 활용하기 위해 새로운 I/O 요청 인터페이스인 Psync IO[3]가 제시되기도 하였다.

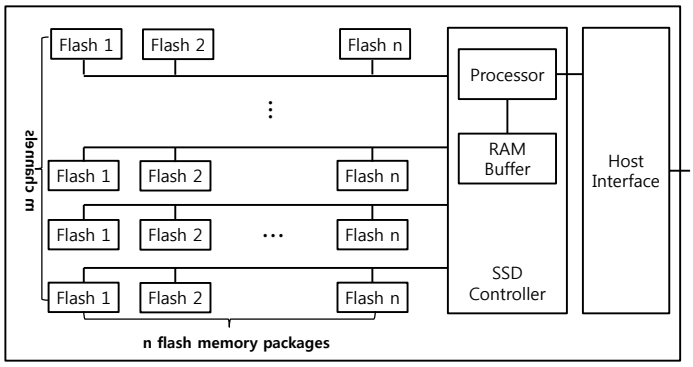



그림 2. 플래시SSD의 내부 구조

하드디스크와 마찬가지로 플래시SSD에도 이중 버퍼링 기법을 적용할 수 있다. 하드디스크는 여러 개의 비동기 I/O 요청을 받아도 데이터의 위치에 따라 디스크 암(disk arm)을 해당 위치로 움직여야하는 단점이 있다. 반면 플래시SSD는 데이터 접근에 물리적인 움직임이 없으므로 오히려 이 방식은 플래시SSD에 더 적합하다고 할 수 있다. 그러나 이중버퍼링 기법에는 한 가지 큰 단점이 있다. 이중버퍼링 알고리즘은 입력버퍼의 어떤 블록이 소모될지 알지 못하기 때문에, 항상 비동기 I/O로 읽어오는 데이터 블록의 수가 n(생성된 런의 개수)으로 고정된다. 따라서 n의 값에 따라 이중버퍼링의 성능이 좌우된다. 만약 n이 너무 작으면 비동기로 읽어오는 데이터 합의 크기가 작아 플래시SSD의 대역폭을 다 사용할 수 없고 하나의 블록이 소모되는 평균 시간이 짧기 때문에 정렬이 멈추게 되는(block) 시간이 커져 성능이 저하된다. 반대로 n이 너무 커서 플래시SSD의 대역폭을 최대로 사용하게 된 시점부터는 그 이상의 블록을 동시에 읽어오는 것이 성능향상에 도움을 주지 않는다. 따라서 본 연구에서는 이러한 문제점을 해결하고 파라미터를 조정하여 성능을 최적화할 수 있는 새로운 합병정렬 알고리즘을 제시한다.

## 2. 새로운 합병정렬 알고리즘

하드디스크에서 합병정렬을 개선하기 위해 중 읽기 스케줄링[4]과 같은 방법이 제안되었다. 읽기 스케줄링은 합병 단계에서 필요한 블록들의 순서인 블록 소모 순서(block consumption sequence - BCS)를 계산한다. 합병 단계에서 블록이 소모되는 시점은 해당 블록의 마지막 레코드가 정렬되는 시점이므로 BCS는 런 생성단계에서 생성된 모든 데이터 블록의 최대키 값을 정렬함으로써 구할 수 있다. 하드디스크에서는 BCS를 디스크 탐색 횟수(Number of disk seek)가 최소화되도록 수정해야 성능을 최대한 이끌어 낼 수 있다. 탐색 횟수가 최소화되도록 BCS를 수정하기 위해서는 모든 경우의 수를 비교해야 하므로 비용이 굉장히 크다. 추론 알고리즘(heuristic algorithm)을 사용하더라도 이에 따른 추가 비용이 발생한다. 반면 플래시SSD는 하드디스크와 달리 여러 개의 비연속적인 블록들을 동시에 읽을 수 있으므로 BCS를 수정할 필요가 없다.

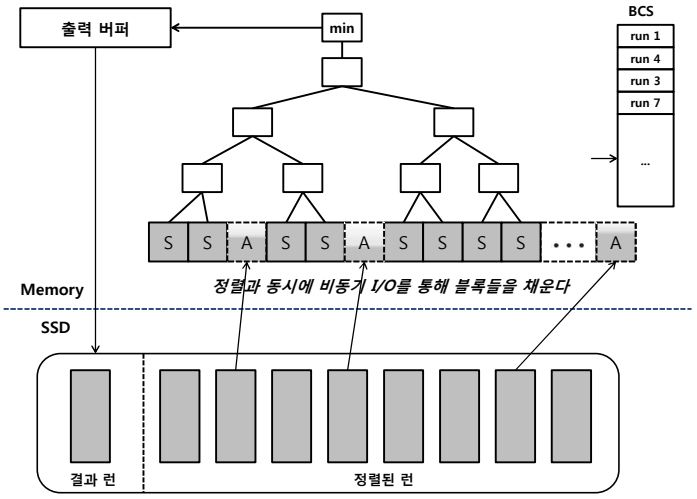



그림 3. 제안하는 합병정렬의 정렬 과정

BCS를 사용하여 합병정렬을 수행하는 과정은 (그림 3)과 같다. 우선 각 런으로부터 하나의 블록을 입력버퍼로 읽어 온다(그림 3의 S블록). S블록들의 레코드들을 트리 구조를 사용해 정렬하는 동안, BCS를 참조하여 다음에 필요한 블록들을 각각 비동기 I/O를 통해 입력버퍼로 읽는다(A 블록). 이때 A블록의 개수는 파라미터에 의해 결정된다. 정렬이 수행되는 과정에서 특정 S블록이 소모되면 A블록 중 하나가 S블록으로 편입되어 정렬에 참여하게 되고 빈 S블록은 A블록이 되어 BCS가 가리키는 다음 블록을 읽게 된다. 이런 과정을 모든 데이터 블록이 소모될 때까지 반복하면 결국 하나의 정렬된 런이 생성된다.

BCS를 사용했을 때 다음과 같은 이점이 있다. 우선 이중 버퍼링과 달리 생성된 런의 개수와 무관하게 비동기로 읽는 블록의 수를 조절할 수 있다. 런의 개수, 한 블록의 크기 등을 고려하여 플래시SSD의 내부 병렬성을 최대화하는 블록의 개수를 계산하고 입력버퍼가 허용하는 한에서 그 개수만큼의 비동기 I/O를 요청하면 최대의 성능을 끌어낼 수 있다. 따라서 이 방식은 런의 개수와 상관없이 항상 최적화된 성능을 보인다. 또한 이중 버퍼링에서는 비동기로 읽는 블록들 중 오랜 시간이 지난 뒤에야 정렬에 참여하는 블록이 생긴다. 이런 블록들은 오랫동안 불필요하게 입력버퍼를 점유한다. 반면 제안하는 알고리즘은 BCS를 계산하여 필요한 순서대로 블록을 읽어오므로, 불필요하게 입력버퍼를 오랫동안 차지하는 블록이 존재하지 않는다.

따라서 본 알고리즘은 이중 버퍼링보다 더 성능이 안정적이고(stability) 더 효율적인 메모리 활용 능력을 갖는다.

## 3. 결 론

본 논문은 플래시SSD에서 사용하기 적합한 새로운 합병 정렬 알고리즘을 제안하고, 이 알고리즘이 기존의 이중 버퍼링보다 어떤 측면에서 우수한지에 대해 서술하였다. 새롭게 제안한 알고리즘은 런 생성 단계에서 블록 소모 순서(BCS)를 계산하고 합병 단계에서 이를 이용해 조만간 필요하게 될 여러 개의 블록을 비동기 I/O로 요청하여 읽는다. 이러한 방식을 사용하면 합병 단계에서 런 개수에 구애받지 않고 필요한 만큼(플래시SSD의 대역폭을 최대한 활용하는)의 블록을 읽을 수 있으므로 성능이 뛰어나다고 알려진 이중 버퍼링보다 안정적이며 공간 활용 능력도 더 뛰어나다.

실질적으로 합병정렬은 DBMS의 조인(sort-merge join), 중복제거(duplication elimination) 등의 핵심 연산 구현에 사용되나[5] 아직까지 대부분의 DBMS에서는 플래시SSD에 적합한 알고리즘이 탑재되어 있지 않다. 따라서 본 논문에서 제시한 알고리즘을 DBMS에 탑재하게 된다면 DBMS 의 성능을 좀 더 개선할 수 있을 것이라 생각된다.

## 참 고 문 헌

[1] S. Dasgupta, C. Papadimitriou, and U. Vazirani, “Algorithms,” 2008.

[2] F. Chen, R. Lee, and X. Zhang, “Essential roles of exploiting internal parallelism of flash memory based solid state drives in high-speed data processing,” In High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on, pages 266-277. IEEE, 2011.

[3] H. Roh, S. Park, S. Kim, M. Shin, and S.-W. Lee, “B+-tree index optimization by exploiting internal parallelism of flash-based solid state drives,“ VLDB 2012.

[4] L. Zheng and P. Larson, “Speeding Up External Mergesort,“ IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 8, NO. 2, APRIL 1996.

[5] R. Elmasri, and S. B.Navathe, “Fundamentals of DATABASE SYSTEMS 5th edition,” 2007.