

논문 2015-10-11

낸드 플래시 메모리 기반의 비대칭 행렬 전치 기법

(An Asymmetry Matrix Transposition Scheme based on NAND Flash Memory)

김 성 철, 박 웅 규, 온 병 원, 이 인 규, 최 규 상*

(Sung-Chul Kim, Woong-Kyu Park, Byung-Won On, Ingyu Lee, Gyu Sang Choi)

Abstract : In this paper, we proposed a new matrix transposition scheme, called asymmetry sub-matrix, and conducted the in-depth performance evaluation of the proposed scheme with other prior schemes, including element-major, row-major and sub-matrix schemes in large-scale matrix. In our results, the proposed asymmetry sub-matrix scheme shows the best performance compared to other prior schemes, while sub-matrix scheme shows the second best performance.

Keywords : Transpose, Matrix, NAND Flash memory, Asymmetry

1. 서 론

NAND 플래시 메모리는 기존의 HDD에 비하여 높은 성능, 낮은 전력 소비, 충격에 대한 내성, 작은 폼팩터 등의 여러 가지 장점들로 인해 임베디드 시스템에 더욱 많이 사용되고 있다. 하지만 NAND 플래시 메모리는 디스크와 다른 독특한 특성이 있기 때문에 플래시 메모리 기반 저장장치 상에서 디스크 기반의 어플리케이션을 곧바로 사용하는 것은 여러 가지 문제점이 발생할 수 있다. 특히 임의 쓰

기와 같은 특정한 작업에서 HDD와 비교하여 비슷한 성능을 보이는데, 이는 NAND 플래시 메모리의 자체의 물리적 특성(쓰기 전 소거 구조 : erase-before-write) 때문에 성능을 높이는 것에는 한계가 있다. 첫째, 지우기 연산은 NAND 플래시 메모리의 성능에 가장 부정적인 영향을 주는 연산으로 블록 단위로 지우기 연산이 이루어지며, 대략 1ms라는 매우 긴 연산 시간이 걸린다. 둘째, NAND 플래시 메모리는 페이지(page) 단위로만 읽기/쓰기를 하여야 하고 지우기 연산은 블록(block) 단위로 처리하여야 한다. 즉 하나의 페이지에 데이터를 기록할 때 해당 페이지가 이미 데이터를 가지고 있다면, 이 페이지를 포함하고 있는 블록을 삭제하여 초기화한 후 데이터를 기록하는 특징이 있다 [1, 2]. 따라서 이러한 구조에서 데이터 접근 시 읽기와 쓰기 연산만을 사용하는 디스크 기반의 어플리케이션을 곧바로 사용하는 것은 비효율적이다. 이러한 문제를 해결하기 위해 플래시 전환계층(flash translation layer, FTL)[3,4,5,6]이라고 불리는 소프트웨어가 개발 되었다. 이것은 NAND 플래시 메모리의 물리적 특성으로 인한 단점을 디스크 기반의 저장장치와 같은 환경으로 만들어주기 위한 드라이버 이다. 플래시 전환계층의 주요 역할은 응용 프로그램이 데이터를 유지하고 있는 페이지에 데이터를 기록하려고 할 때 비어있는 페이지에 데이터를 기록하도록 주소를 재지정 해주는 것이다. 그러나 플래시 전환계층을 통해 반복적인 쓰기 연산이

*Corresponding Author(castchoi@ynu.ac.kr)

Received: 22. Oct. 2014, Revised: 5 Jan. 2015,

Accepted: 12. Jan. 2015.

S.-C. Kim, W.-K. Park, G.S. Choi: Yeungnam University

B.-W. On: Kunsan National University

I. Lee: Troy University, USA

※이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 일반연구자지원사업 지원을 받아 수행된 것임 (No. 2013012524)

※본 연구는 2013년도 산업통산자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다.(No. 20132010101800)

※이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2013R1A1A2063304)

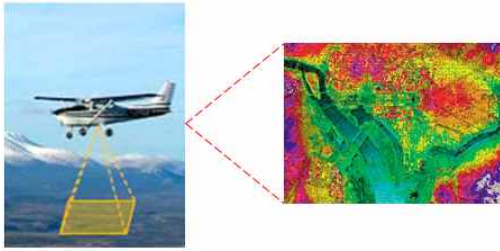


그림 1. 합성 개구 레이더

Fig. 1 SAR : Synthetic Aperture Radar

발생했을 때 논리적인 주소를 물리적인 주소로 전환시킴으로써 자기 디스크와 같은 환경을 제공해 줄 수 있지만 많은 제자리갱신 연산(overwrite)이 발생했을 경우 성능이 저하되는 것을 피할 수 없다.

2차원 디지털 필터 또는 2차원 Fast Fourier Transforms(2D-FFT)등을 하기 위해서는 처리해야 할 데이터가 방대하기 때문에 필수적으로 이차원 행렬의 전치를 필요로 한다. 그림 1과 같이 항공기나 인공위성으로 광범위한 지역에 걸쳐 관측한 레이더 정보를 수학적으로 재구성해 영상화 하는 기술인 합성 개구 레이더(SAR) 또는 무인항공기(UAV)의 경우 2차원 디지털 필터와 2D-FFT로 처리해야 할 데이터가 방대하며 실시간 처리를 요구하므로 데이터 행렬의 전치가 필수적이며 전치 기법의 성능이 그 시스템의 성능에 큰 영향을 미치게 된다. 또한 그림 1과 같이 기상청의 기상예보 데이터의 경우 방대한 행렬로 구성되어 있으며 그 행렬의 곱셈 연산이 빈번히 발생하여 전치행렬 기법이 많이 사용 된다.

II. 기존의 행렬전치 기법

현재의 전치행렬 기법은 크게 세 가지 형태이다. 하나는 행렬의 각 원소 단위로 전치하는 것이며, 다른 하나는 하나의 열 또는 행 단위로 전치하는 것이다. 마지막으로 행렬을 부분행렬로 나누어 전치하는 방법이 있다. 첫 번째로 각 원소 단위로 전치하는 기법의 경우 각 원소에 대하여 개별적으로 접근을 하는 방식으로서 너무 많은 접근이 필요하여 시간이 오래 걸리는 단점이 있다. 두 번째로 하나의 행 또는 열 단위로 전치하는 기법은 한번에 하나의 행 또는 열을 읽어와서 전치 후 저장하므로 첫 번째의 각 원소 단위로 전치하는 기법에 비하여 메인 메모리의 활용도가 증가하고 읽기와 쓰기의 횟수를

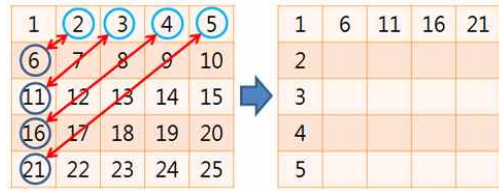


그림 2. 원소 단위로 행렬 전치하는 기법

Fig. 2 An example of scheme using element unit

줄일 수 있어 전체적인 성능을 향상시킬 수 있다. 하지만 이 기법의 경우 한 개의 행 또는 열이 메인 메모리에 한번에 읽어 올 수 없을 경우 사용할 수 없는 단점이 있다. 마지막으로 부분행렬단위로 전치하는 기법은 원래의 행렬을 작은 부분행렬로 나누어 전치하는 기법으로 위의 두 번째 기법의 단점인 메인 메모리 크기에 따른 종속성을 해결하여 메인 메모리의 크기에 관계없이 사용할 수 있는 장점이 있다.

행렬을 전치하는 가장 간단한 기법은 행렬을 모두 한꺼번에 읽어 각 원소끼리 전치하는 기법이다. 위 설명과 같이 그림 2는 각 원소끼리 전치하는 원소단위 전치 기법이다. 첫 번째 열의 각 원소들을 전치하여 대응되는 첫 번째 행에 저장하는 방식이다. 이 기법은 가장 단순하지만 만약 행렬의 크기가 커서 메인 메모리에서 한 번에 다 저장하지 못하게 된다면 사용할 수가 없게 된다. 다른 방법으로는 각 원소를 하나씩 메인 메모리로 읽어와 전치하는 방법이 있는데 이 또한 너무 많은 읽기와 쓰기 명령이 필요하여 실시간 처리가 필요한 임베디드 시스템에서는 사용하기가 힘들다. 특히 NAND 플래시 메모리의 경우 읽기와 쓰기를 하는 단위인 페이지 단위로 데이터를 읽어 오는데 각 원소단위가 페이지 크기보다 작기 때문에 읽어온 데이터의 대부분은 사용하지 않아 불필요하게 된다. 또한 다른 원소를 메모리로 읽어올 때 이전에 읽어온 페이지를 다시 읽어 와 그 중 필요한 데이터만 사용한다. 따라서 이전에 읽었던 페이지를 다시 읽어 그 중 하나의 데이터만 사용하게 되므로 아주 비효율적이고 많은 실행시간을 필요로 한다. 본 논문에서는 기상예보 시스템, 합성 개구 레이더 또는 무인항공기와 같이 대형의 행렬을 이용하는 임베디드 시스템을 위한 최적의 기법을 찾고자 하였기 때문에 메인 메모리에 한 번에 다 저장하지 못하는 행렬을 가정하였다. 따라서 메인 메모리에서의 연산시간 이외에 NAND 플래시 메모리에서의 읽기와 쓰기 시간 또

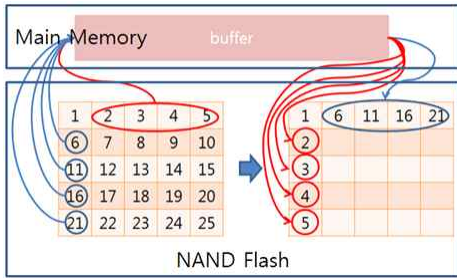


그림 3. 행 단위로 전치하는 기법
 Fig. 3 An example of scheme using row unit

한 중요하게 되고 원소단위로 행렬 전치 기법은 실시간으로 사용하기 부적절하다.

원소단위로 행렬 전치 기법의 단점을 고려한 다른 기법은 행렬의 행과 열을 각 라인 별로 메인 메모리에 읽어 사용하는 방법이다. 이 방법은 이전의 기법에 비하여 하나의 행 또는 열만을 메인 메모리에 읽어 들여 사용하는 방법이므로 원소단위로 전치하는 기법에 비하여 메인 메모리가 작더라도 사용이 가능한 기법이다. 이 기법은 앞서 언급했던 행렬을 전치하여 새로운 행렬로 저장할 경우 읽기와 쓰기 하는 방식에 따라 두 가지 방법으로 나눌 수가 있다. 첫째는 하나의 행을 한꺼번에 읽어 각각의 원소들을 열로 쓰는 것이고 둘째는 하나의 열에 있는 각 원소들을 읽어 행으로 한꺼번에 쓰는 것이다. 첫 번째 방법은 행이 연속된 데이터 이므로 한 번에 읽을 수 있으므로 읽기시간이 적게 걸리지만 원소들을 각 열에 써줘야 하므로 쓰기 시간이 오래 걸리게 된다. 두 번째 방법은 열이 떨어져 있는 데이터이므로 각 원소들을 읽어오는 시간은 오래 걸리지만 행에 한 번에 쓸 수 있으므로 쓰기 시간이 짧게 된다. 열 단위로 전치하는 기법의 경우 그림 3의 푸른색 선과 같이 한 열의 원소들을 저장장치에서 메인 메모리로 각각 읽어 저장한다. 그 후 메인 메모리에 있는 열의 데이터들을 전치하여 하나의 행으로 만든 후 저장장치에 연속되는 데이터로 쓰기 명령을 한다. 이 경우 하나의 열에 있는 원소들을 각각 읽어야 하므로 저장장치로의 다섯 번의 읽기 명령이 필요하다. 그리고 전치 후 연속된 원소들로 한 번의 쓰기 명령이 필요하다. 행 단위로 전치하는 기법의 경우 그림 3의 붉은색 선과 같이 저장장치에 있는 하나의 행의 모든 원소를 메인 메모리로 읽어온다. 그 후 전치하여 각 열에 대한 주소에 맞게 원소를 각각 쓰기 명령을 한다. 따라서 그림 3의

행 단위로 전치하는 기법의 경우 하나의 행에 대해 연속된 원소들을 한 번에 읽어오는 명령과 전치하여 각 열에 저장하는 다섯 번의 쓰기 명령이 필요하다. 이 기법들은 읽기와 쓰기 시간이 다른 저장장치들에서 선택적으로 사용할 수 있다. 특히 NAND 플래시 메모리와 같이 읽기보다 쓰기 시간이 긴 저장장치의 경우 두 번째 기법이 좀 더 유리한 것을 알 수 있다. 즉 NAND 플래시 메모리의 경우 다섯 번의 읽기, 한 번의 쓰기 명령이 필요한 열 단위로 전치하는 기법 보다 한 번의 읽기, 다섯 번의 쓰기 명령이 필요한 행 단위로 전치하는 기법이 성능이 더 저하 된다. 하지만 이렇게 선택적으로 사용하는 것은 행렬을 전치하여 새로운 행렬로 저장할 경우에 대해서만 적용하여 사용할 수 있고 본 논문에서 초점을 맞춘 행렬을 전치하여 보조 기억장치의 기존의 위치에 덮어쓰기 할 경우에 대해서는 대응되는 데이터들을 동시에 전치해야 하므로 그림 3에서 붉은색 선과 푸른색 선인 두 가지 방법을 같이 쓰는 것과 동일한 효과를 가진다.

이 기법들의 경우에도 하나의 행 또는 열이 메인 메모리에 모두 저장할 수 없는 크기일 경우 사용할 수 있지만 여러 번 나누어 전치해야 하고 마지막 페이지에서 사용하지 않는 데이터가 많이 남을 수 있다는 단점이 있다. 또한 행 또는 열이 메인 메모리에 모두 저장할 수 있더라도 그 크기에 따라 메인 메모리를 사용하지 않는 부분이 발생하여 비효율적으로 연산을 해야 하는 경우가 발생하게 된다.

세 번째로 부분행렬 단위로 전치하는 기법의 경우 앞서 언급한 두 가지 기법에 비하여 버퍼의 크기에 대한 종속성이 줄어들어 사용이 효율적이지만 이 또한 대각행렬에 포함되는 부분행렬의 경우 버퍼의 반만 사용하기 때문에 충분히 효율적이지 못하다[2]. 그림 4는 부분행렬을 이용하여 전치하는 기법으로 왼쪽의 기존 행렬을 작은 부분행렬로 나누어 버퍼로 옮겨 각 부분행렬의 내부를 전치한 다음 전체행렬에서 부분행렬들을 전치된 위치로 저장하는 방식이다. 이 기법의 경우 전치단위를 부분행렬을 사용하여 기존의 기법에 비하여 버퍼에 크기에 대한 제약이 사라지게 되고 기법을 진행함에 따라 전치하는 데이터의 수가 동일하여 버퍼를 효율적으로 사용한다. 하지만 이 기법은 모든 부분행렬의 크기가 동일함으로 그림에서 두 번째와 세 번째 부분행렬과 같이 전체 버퍼를 모두 사용하는 것과 달리 첫 번째 부분행렬이나 네 번째 부분행렬과 같이 대각행렬에 해당하는 부분행렬의 경우 전체 버

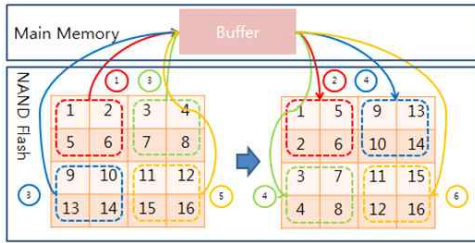


그림 4. 부분행렬을 이용한 전치 행렬 기법
Fig. 4 An example of scheme using sub-matrix unit

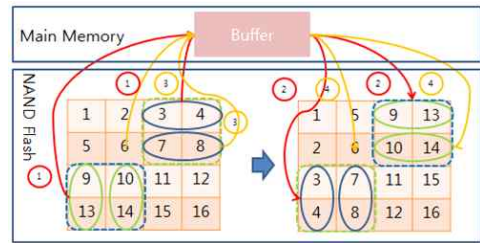


그림 5. 비대칭 부분행렬을 이용한 전치 기법
Fig. 5 An example of scheme using asymmetry matrix unit

퍼의 반만을 사용하여 전치하여야 함으로 좀 더 발전시킬 수 있는 여지가 있다.

그림 4는 부분행렬을 이용한 행렬 전치 기법을 설명한 것이다. 우선 행렬을 버퍼의 크기에 맞게 작은 부분행렬로 나누어 준다. 작게 나눈 부분행렬 중 붉은색 첫 번째 행렬을 메모리의 버퍼로 읽어와 전치하여 다시 NAND 플래시 메모리의 원래 행렬이 있던 주소에 저장한다①②. 다음 녹색 부분행렬과 푸른색 부분행렬을 메인 메모리의 버퍼로 읽는데 이 두 부분행렬은 전치하여 서로의 위치가 바뀌게 되므로 동시에 메인 메모리의 버퍼에 저장할 수 있어야 한다③. 녹색과 푸른색 부분행렬을 메인 메모리의 버퍼에서 전치하여 NAND 플래시 메모리에 위치를 바꾸어 저장한다④. 나머지 주황색 부분행렬도 첫 번째 붉은색의 부분행렬과 같이 메인 메모리의 버퍼로 읽어 전치한 후 NAND 플래시 메모리의 원래 주소에 저장한다⑤⑥. 이 과정과 같이 대각행렬에 해당되는 부분행렬과 대각행렬에 해당되지 않는 부분행렬에 대하여 처리하는 과정이 조금 다르다. 대각행렬에 해당되는 부분행렬은 전치하여도 NAND 플래시 메모리에서 같은 위치에 저장되기 때문에 각각 처리할 수 있지만 대각행렬에 해당되지 않는 부분행렬에 경우 전치하게 되면 NAND 플래시 메모리에서 저장되는 위치가 바뀌기 때문에 그 대응되는 부분행렬과 함께 처리를 해주어야 한다.

III. 비대칭 부분행렬을 이용한 행렬 전치 기법

본 논문에서 제안하는 기법은 부분행렬을 이용한 것으로 기존의 부분행렬을 이용한 기법과 다르게 대각행렬에 해당하는 부분행렬은 같지만 나머지

부분행렬에 대하여 연산과정이 다르다. 기존의 부분행렬을 이용한 기법은 부분행렬의 크기가 버퍼의 전체 크기에 절반으로 설정하지만 본 논문에서 제안하는 기법의 부분행렬의 크기는 버퍼의 크기와 동일하다. 대각행렬에 해당되는 부분행렬은 기존의 부분행렬을 이용한 기법과 동일하게 연산되지만 대각행렬에 해당되지 않는 부분행렬의 경우 기존의 부분행렬을 이용한 기법과 달리 다시 나누어 연산한다. 본 논문에서 제안하는 기법은 기존의 부분행렬 단위로 전치하는 기법의 문제점들을 해결하기 위해 이루어진 것으로서, 버퍼의 크기에 종속적이지 않고 읽기와 쓰기명령의 횟수를 최적화하여 전치의 성능을 향상시키도록 한 기법이다. 각 부분행렬들은 서로 대응되는 부분행렬을 제외하고는 독립적으로 전치가 되므로 부분행렬의 모양에 상관없이 전치된 후 전체행렬을 만들게 되면 기존의 전체행렬을 나누지 않고 전치한 것과 같은 결과가 나오게 된다. 따라서 기존의 기법에서 하나의 행 또는 열 단위로 나누어 전치하는 기법이 버퍼의 크기에 따라 불가능한 경우가 발생할 수 있는 것에 반해 부분행렬로 나누어 전치하는 기법은 버퍼의 크기에 맞게 부분행렬의 크기를 조절할 수 있기 때문에 버퍼의 크기에 종속적이지 않다. 그리고 기존의 부분행렬을 단위로 전치하는 기법에서 대각행렬에 해당되는 부분행렬의 경우 버퍼를 절반만 사용할 수 있는 것에 비하여 본 논문에서 제안한 기법은 이 경우에도 버퍼를 모두 사용할 수 있는 장점이 있다. 또한 NAND 플래시 메모리와 같이 읽기와 쓰기 지우기를 하는 단위가 지정되어 있는 경우에 대해서도 행 또는 열 단위로 전치하는 기법에 비하여 부분행렬의 크기를 조절하여 가장 적절한 크기로 나누어 최대한의 성능을 발휘할 수 있다.

그림 5는 본 논문에서 제안하는 기법으로 대각행렬에 해당하는 부분행렬은 기존의 기법과 동일하

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

그림 6. 비대칭 부분행렬을 이용한 전치 기법의 행렬

Fig. 6 An example of matrix using asymmetry matrix unit

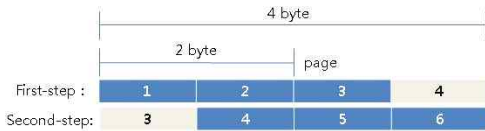


그림 7. 메인 메모리에서의 데이터(비대칭 부분행렬을 이용한 행렬 전치 기법)

Fig. 7 Data organization of main memory (scheme using asymmetry matrix unit)

나 대각행렬이 아닌 부분행렬에 대해서는 다시 부분행렬을 반으로 나누어 각각에 대하여 전치한다. 기존의 기법과 가장 큰 차이점은 위 그림과 같이 함으로 인하여 부분행렬의 크기가 기존의 기법에 비하여 2배가 커지게 되어 페이지의 전체 읽기와 쓰기 수가 줄어들 수 있다는 것이다. 주의할 점은 읽기와 쓰기 명령의 수는 부분행렬을 이용한 행렬 전치 기법에 비하여 도리어 커지게 된다. 이유로는 그림 5에서 푸른색 부분행렬의 경우 부분행렬을 이용한 행렬 전치 기법은 데이터 9,10이 연속된 데이터 이므로 한번에 읽어 메인 메모리에 저장하고 데이터 13,14또한 연속된 데이터 이므로 한번에 읽어 메인 메모리에 저장 하지만 비대칭 부분행렬 행렬 전치 기법을 사용할 경우 데이터 9와 13을 읽어야 하므로 데이터가 연속되지 않아 각각 읽어야 하므로 읽기 명령의 횟수가 부분행렬을 이용한 행렬 전치 기법에 비하여 증가 하였다. 그림 6의 행렬을 비대칭 부분행렬을 이용한 전치 기법을 이용하여 전치할 때 처음 푸른 부분행렬 부분을 예로 사용하여 설명한다. 여기서 페이지의 단위는 2 byte이다. 처음 연속된 데이터 1, 2, 3을 메인 메모리의 버퍼로

읽기 위해서는 페이지의 단위가 2 byte이므로 데이터 1, 2까지는 하나의 페이지로 읽을 수 있지만 데이터 3을 읽기 위해서는 데이터3이 포함된 하나의 페이지를 더 읽어 와야 한다. 즉, 메인 메모리에는 1, 2, 3, 4가 읽어지게 되는데 그림 7의 첫 번째와 같이 메인 메모리에 저장이 된다. 그 중 실제로 사용되는 데이터는 1, 2, 3이므로 4는 쓸모가 없는 데이터이다. 따라서 비효율적으로 보이지만 다음 붉은색 부분행렬을 전치할 때 이전에 읽어왔던 데이터 4가 필요로 하게 된다. 즉, 푸른색 부분행렬만을 볼 때는 비효율적이지만 붉은색 부분행렬과 함께 본다면 이전에 읽어왔던 페이지를 재사용할 수 있어 비효율적이지 않게 되는 것이다. 마찬가지로 녹색 부분행렬의 경우도 동일하게 적용된다. 따라서 그림 7에서 읽는 페이지의 수는 3페이지이지만 처음 2페이지를 연속으로 읽으므로 페이지를 읽는 횟수는 2번이다.

앞서 예시로서 설명한 비대칭 부분행렬단위 전치는 부분행렬을 다시 반으로 나누어 전치하는데 이는 부분행렬의 한 행 길이가 소수일 경우 나눌 수가 없다는 문제점이 생긴다. 또한 기존의 부분행렬단위 전치 기법에서도 전체 행렬의 한 행 길이가 소수일 경우 동일한 문제가 발생할 수 있다. 이 경우 본 논문에서 해결했던 방법은 전체 행렬을 나눌 수 있는 최대한의 크기의 부분행렬로 나누어 부분행렬 또는 비대칭 부분행렬 전치 기법을 사용하여 전치하며 나머지 부분에 대해서는 행과 열 단위로 행렬 전치 기법을 사용하여 전치하는 방법을 사용하였다.

예를 들어 7X7행렬의 경우 한 행의 길이가 소수이므로 반으로 나눌 수 없다. 이 경우 전체행렬에서 나눌 수 있는 최대한의 크기의 부분행렬인 6X6행렬로 나누어 부분행렬 전치 기법을 사용하여 전치하며 비대칭 부분행렬 전치 기법에서도 동일하게 부분행렬의 행 또는 열의 길이가 소수일 경우 나눌 수 있는 최대한의 크기의 부분행렬로 나누어 비대칭 부분행렬 전치 기법을 사용하여 전치하고 나머지 부분에 대해서 행 또는 열 단위 행렬 전치기법을 사용하여 전치한다. NxM 과 같이 정방행렬이 아닐 경우에도 NxN의 정방행렬과 마찬가지로 버퍼의 크기를 고려하여 가장 큰 정방부분행렬로 나누어 비대칭 부분행렬 전치기법으로 전치한 뒤 정방부분행렬에 포함되지 않는 부분을 행 또는 열 단위 행렬 전치기법을 사용하여 전치한다.

표 1. 실험 환경

Table 1. Experiment environment

CPU	Intel Core2 Duo 3.0GHz
Main Memory	1.9Gib
Storage Interface	Serial-ATA-2
File System	ext4
OS	Linux fedora 2.6.35.6
SSD	Intel SSD330 240GB (buffer 32MB)
sequential read	500 MB/s
sequential write	450 MB/s
random read(8GB span)	22500 IOPS
random write(8GB span)	33000 IOPS

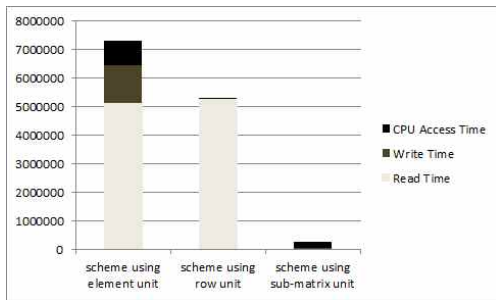


그림 8. NAND 플래시 메모리에서 8GB 행렬의 각 기법에 대한 전치 성능(256KB 버퍼사용)

Fig. 8 Performance comparison for scheme in 8GB NAND flash memory (using 256KB buffer)

IV. 실험 결과 및 분석

본 장에서는 제안 기법과 기존 기법의 성능을 비교 분석하기 위해서 실제 NAND 플래시 메모리와 HDD환경에서 행렬전치 알고리즘을 표 1에 나와 있는 환경에서 실험해보고 각각의 성능에 대하여 분석한 후 결과를 도출한다.

표 1은 실험에 사용하였던 시스템과 SSD의 환경이다.

그림 8은 8GB 행렬을 전치 할 때 256KB단위 별로 전치에 대한 성능을 실험 한 것이다. 읽기와 쓰기 횟수는 원소 단위 행렬 전치 기법이 가장 많고 열 단위 행렬 전치 기법은 읽기 횟수는 동일 하지만 쓰기 횟수를 많이 줄일 수 있다. 부분행렬 단 행렬 전치 기법의 경우 읽기 횟수는 줄 수 있지만

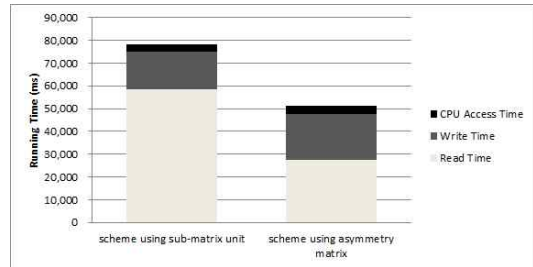


그림 9. NAND 플래시 메모리에서 2GB크기의 행렬에 대한 기존 부분행렬을 이용한 기법과 본 논문에서 제안하는 기법의 성능(256KB 버퍼사용)

Fig. 9 Performance comparison for scheme using sub matrix and asymmetry matrix in 2GB NAND flash memory (using 256KB buffer)

쓰기 횟수는 열 단위 행렬 전치 기법에 비하여 쓰기 횟수는 조금 늘어 난다. 앞의 결과에서 읽기 연산에 소모되는 시간 보다 쓰기 연산에 소모되는 시간이 더 짧은 것을 볼 수 있다. 이것은 NAND 플래시 기반의 저장장치인 SSD는 쓰기 연산이 읽기 연산에 비해 소모되는 시간이 크고, 추가적인 쓰기 및 읽기 연산이 발생할 수 있으며, 쓰기 연산으로 발생할 수 있는 삭제 연산은 장치의 수명을 단축시킨다는 단점을 해소하기 위해 쓰기 버퍼를 사용하고 있기 때문이다[7]. 기존의 행렬 전치 기법을 실제 NAND 플래시 메모리에서 실험하였을 때 이론에서 밝혔던 바와 같이 부분행렬을 이용한 행렬 전치 기법, 행 단위 행렬 전치 기법, 원소 단위 행렬 전치 기법 순으로 성능이 좋은 것을 알 수 있었다. 앞으로는 기존의 행렬 전치 기법 중 가장 성능이 좋은 부분행렬을 이용한 행렬 전치 알고리즘과 본 논문에서 제안하고 있는 비대칭 부분행렬 알고리즘에 대하여 실험하고 비교한다.

그림 9는 기존의 부분행렬을 이용한 기법과 본 논문에서 제안하는 기법의 성능을 비교한 것이다. 전체적인 성능은 약 1.5 배 향상된 것을 볼 수 있다. 이것은 부분행렬의 크기가 달라짐에 따라 읽기와 쓰기명령의 수가 전체적으로 줄어들게 되어 속도가 향상되는 것을 알 수 있다. 즉 기존의 부분행렬기법에 비하여 제안하는 기법은 사용할 수 있는 부분행렬의 크기가 커져 버퍼의 남은 부분이 적도록 효율적으로 사용하여 성능이 향상된다.

그림 10은 다양한 크기의 버퍼에 대하여 부분행렬 전치 기법과 비대칭 부분행렬 전치 기법간의 성

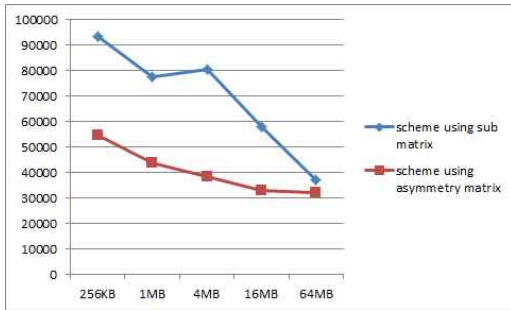


그림 10. 다양한 크기의 버퍼에 대한 부분행렬 전치 기법과 비대칭 부분행렬 전치 기법의 실행시간

Fig. 10 execution time comparison for asymmetry and sub matrix scheme with various buffer size

능을 비교한 그림이다. 비대칭 부분행렬 전치 기법의 경우 부분 행렬 전치 기법에 비하여 작은 버퍼를 사용하였을 시 좀 더 유리한 성능을 나타내는 것을 볼 수 있다. 이것은 페이지의 읽기 횟수가 버퍼가 커짐으로 인해 그 횟수의 차이가 줄어들어 점차 성능의 차이 또한 비슷해지는 결과를 나타내는 것이다.

V. 결 론

이 논문에서는 부분행렬의 모양과 크기를 조절하여 최적의 버퍼 사용과 읽기, 쓰기명령의 수를 최적화 하는 것에 대하여 분석하고 연구하였다. 이전의 행렬 전치 기법은 일정 이상의 메인 메모리의 버퍼를 요구하여 메인 메모리의 버퍼가 클수록 좋은 성능을 가진다. 하지만 제안하는 기법인 비대칭 부분행렬을 이용한 행렬 전치 알고리즘의 경우 적은 메인 메모리의 버퍼로도 기존의 알고리즘에 비하여 좋은 성능을 가질 수 있음을 실험을 통하여 보였다. 따라서 비대칭 부분행렬을 이용한 행렬 전치 기법을 사용하여 현재 행렬 전치 기법을 사용하는 삼성 개구 레이더와 같이 임베디드 시스템에서 제한된 자원을 가질 때 기존의 행렬 전치 기법에 비하여 좋은 성능을 발휘하여 좀 더 저가의 고성능 임베디드 시스템을 구축할 수 있을 것으로 보인다.

References

- [1] B.-S. Jung, J.-H. Lee, "Analysis on the effectiveness of the filter buffer for low power NAND flash memory," IEMEK J. Embed. Sys. Appl., Vol. 7, No. 4, pp. 201-207, 2012 (in Korean).
- [2] J. Ryu, C. Park, "A technique to enhance performance of log-based flash memory file systems," IEMEK J. Embed. Sys. Appl., Vol. 2, No. 3, pp. 184-193, 2007 (in Korean).
- [3] J. Kim, J.M. Kim, S.H. Noh, S. Min, Y. Cho, "A space-efficient flash translation layer for compact flash systems," IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, pp. 366-375, 2002.
- [4] T.-S. Chung, D.-J. Park, S. Park, D.-H. Lee, S.-W. Lee, H.-J. Song, "A survey of flash translation layer," Journal of Systems Architecture, Vol. 55, No. 5-6, pp. 332-343, 2009.
- [5] S.-W. Lee, D.-J. Park, T.-S. Chung, D.-H. Lee, S. Park, H.-J. Song, "A log buffer based flash translation layer using fully associative sector translation," ACM Transactions on Embedded Computing Systems, Vol. 6, No. 3, pp. 436-453, 2007.
- [6] H.-S. Lee, H.-S. Yun, D.-H. Lee, "HFTL: hybrid flash translation layer based on hot data identification for flash memory," IEEE Transactions on Consumer Electronics, Vol. 55, No. 4, pp. 2005-2011, 2009.
- [7] S. Kim, T. Kim, "A write buffer management scheme considering the command queue in SSD," Journal of KIISE Vol. 39 No. 1A, pp. 313-315, 2012 (in Korean).
- [8] Samsung Electronics Company, PM810 Data Sheet, 2011.
- [9] Samsung Electronics Company, K9PDG08U5D 128G bit*8 bit NAND Flash-Memory Data Sheet, 2011.

Sung-Chul Kim (김 성 철)

He received B.S. degree in department of information and communication engineering from Yeungnam University, Korea, in 2012. He received M.S. degree in department of information and communication engineering from Yeungnam University, Korea, in 2014. His research Interests Non-volatile memory, Data base.

Email : borame30@ynu.ac.kr

Woong-Kyu Park (박 응 규)

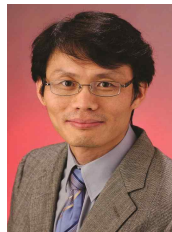
He received B.S. degree in department of information and communication engineering from Yeungnam University, Korea, in 2012. He received M.S. degree in department of information and communication engineering from Yeungnam University, Korea, in 2014. His research Interests Data base, Big data

Email: pwkyu@ynu.ac.kr

Byung-Won On (온 병 원)

He received Ph.D. degree in department of computer engineering from Pennsylvania State University, in 2007. Currently, he is professor in the department of statistics computer science in Kunsan National University, Korea. His research Interests Data mining, Data base, information retrieval, Big data

Email : bwon@kunsan.ac.kr

Ingyu Lee (이 인 규)

He received Ph.D. degree in department of computer engineering from Pennsylvania State University, in 2007. From 2007 to 2013, he was professor in the Troy University, United States. Currently, he is professor in the Troy University, United States. His research Interests Data mining, Social network analysis and business analytic

Email : inlee@troy.edu

Gyu Sang Choi (최규상)



He received Ph.D. degree in department of computer engineering from Pennsylvania State University, in 2005. Currently, he is professor in the department

of information and communication engineering from Yeungnam University, Korea. His research Interests Non-volatile memory, Operating system, Storage system

Email : castchoi@ynu.ac.kr