GPUs 환경에서의 비정형 행렬-행렬 곱셈 연산을 위한 NVIDIA cuBLAS 서브-루틴 성능 분석

김인서, 김동현, 김진성

중앙대학교

inseo764@cau.ac.kr, jeus5771@cau.ac.kr, kimjsung@cau.ac.kr

Performance Analysis of NVIDIA cuBLAS Sub-Routines for Irregular-shaped Matrix-Matrix Multiplication on GPUs

Inseo Kim, Donghyun Kim, Jinsung Kim

Chung-Ang Univ.

요 약

빅데이터와 인공지능은 4차 산업혁명의 핵심기술로, 이들은 서로 밀접하게 연결되어 있다. 높은 성능의 인공지능 개발을 위해서는 고성능컴퓨팅 환경에서 빅데이터를 활용하는 것이 중요하다. 이러한 환경에서, GEMM 연산은 여러 분야에서 기본적으로 사용되는 연산이다. 계산 과학에서는 대체로 큰 정사각형 행렬을 사용하지만, 데이터 과학과 인공지능 분야에서는 좁은 사각형 형태의 비정형 행렬을 사용하는데, 이는 정사각형 행렬보다 처리 속도가 느리다. 이 느린 속도는 메모리 접근 패턴과 병렬처리 과정의 로드-밸런싱 문제 때문이다. 본 논문에서는 NVIDIA의 cuBLAS 라이브러리를 사용하여 정사각형과 비정형 행렬의 GEMM 연산 성능을 비교 분석한다.

Ⅰ. 서 론

대량의 정형 데이터뿐만 아니라 비정형 데이터로부터 가치를 추출하고 결과를 분석하는 기술인 빅데이터(Bigdata)는 인공지능(AI) 등 과 함께 4차 산업혁명의 핵심기술이며 이 기술들은 서로 밀접하게 연관되어 있다[1][2]. 특히, 향상된 학습률로 인한 높은 성능의 인공지능을 위해 고성능 컴퓨팅(High-Performance Computing) 환경에서 빅데이터를 활용하는 것이 중요하다[3]. 고성능컴퓨팅 환경에서 GEMM(GEneral Matrix-Matrix multiplication) 연산은 데이터 과학, 계산 과학, 기계 학습, 인공지능 등 다양한 분야에서 사용되는 기본 연산이다[4][5][6].

계산 과학 분야에서는 일반적으로 대규모의 정사각형에 가까운 행렬-행렬 곱셈 연산을 수행하지만, 데이터 과학, 기계 학습, 인공지능 등의 분야에서는 좁은 사각형(narrow rectangles)의 형태의 행렬들을 기반으로 하는 비정형 행렬-행렬 곱셈 연산을 수행한다. 하지만 이러한 좁은 사각형 형태의 행렬을 기반으로 하는 행렬-행렬 곱셈 연산은 정사각형 형태의 행렬을 기반을 두는 행렬-행렬 곱셈 연산보다 훨씬 느린 경향을 보인다. 이는 좁은 사각형 형태의 행렬에 대한 메모리 접근 패턴과 병렬처리 과정에서 로드-밸런싱의 부재 등 때문이다. 따라서, 비정형 행렬에 대한 GEMM 연산의 성능 분석은 GPU 기반의 고성능컴퓨팅 환경에서 대량의 데이터를 효율적으로 처리하는 데 있어 중요하다.

본 논문에서는 먼저 정사각형 형태의 행렬을 기반으로 둔 행렬-행렬 곱셈 연산 성능을 행렬의 크기에 따라 수행하여, cuBLAS에서 제공하는 GEMM 연산의 일반적인 성능을 파악한다. 그다음, 비정형 행렬에 대한 GEMM 연산의 예제를 나열하고, 정사각형 형태의 행렬을 기반으로 둔 행렬-행렬 곱셈 연산과의 성능 비교를 위해, 각각 나열된 예제에 대하여 유사한 연산량 갖는 좁은 사각형 형태기반 행렬-행렬 곱셈의 실험 사례에 대한 실험 결과를 분석한다.

본 논문에서는 NVIDIA cuBLAS 라이브러리를 소개하고 비정형 행렬-행렬 곱셈에 관해 설명한다. 또한, 비정형-행렬의 모양에 따른 cuBLAS 라이브러리의 성능을 비교분석 한다.

Ⅱ. 본론

1. NVIDIA cuBLAS 라이브러리와 GEMM 연산

cuBLAS 라이브러리 (the CUDA Basic Linear Algebra Subroutine library)는 NVIDIA CUDA 런타임에서 수행되는 기본 선형대수 서브프로그램(BLAS: Basic Linear Algebra Subprograms)으로 NVIDIA GPU의 계산-자원을 효율적으로 사용한다[7]. 게다가, cuBLAS는 벡터-벡터의 일반화된 연산(AXPY)에 대한 Level-1 연산, 행렬-벡터의 일반화된 연산(GEMV: GEneral Matrix Vector multiplication)에 대한 Level-2 연산, 그리고 행렬-행렬의 일반화된 연산(GEMM)에 대한 Level-3연산을 제공한다. cuBLAS Level-3는 GEMM 연산을 위해서 cublas<t>gemm() 함수를 제공하고 아래 식을 지원한다.

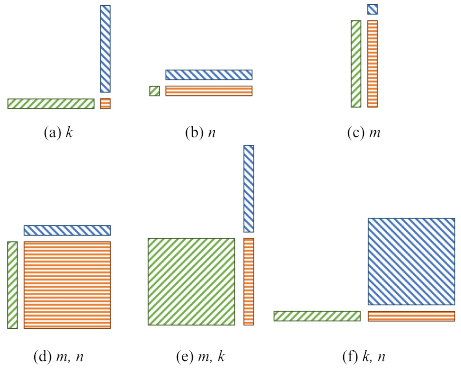
$$C = \alpha op(A) op(B) + \beta C$$

여기서 α와 β는 스칼라 값이고 $op(A)$는 $m \times k$ 크기의 밀집 행렬, $op(B)$는 $k \times n$ 크기의 밀집 행렬, 그리고 C는 $m \times n$크기의 밀집 행렬이다. 본 연구에서는 $op(A)$와$op(B)$에 대해서 모두 전치가 없는 경우만 고려되었다.

2. 실험 결과

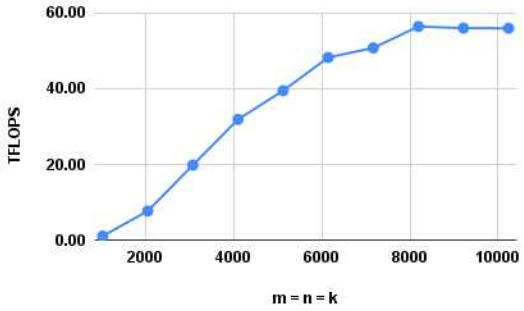
본 연구에서 사용된 실험 환경은 NVIDIA GeForce RTX 4090을 사용했다. 사용된 CUDA 버전은 12.3이고 드라이버 버전은 545.23.08이다. 행렬-행렬-곱셈 연산의 크기와 행렬의 모양 및 크기는 m, n, k에 의해서 결정된다. 그리고 사용된 자료형은 float 타입(4 bytes)이다. 본 연구에서는, [그림 1]에서 볼 수 있듯이, 좁은 사각형 형태의 행렬들을 고려하기 위해, m,

n, k 중 하나의 크기만 큰 경우(3가지: (a), (b) ,(c))와 두 개의 크기만 큰 경우(3가지: (d), (e), (f))를 고려한다.



〔그림 1〕 좁은 사각형 형태들의 예제

아래 [그림 2]은 정사각행렬($m = n = k$)의 크기에 따른 cublassSgemm()함수의 성능을 보여준다, 행렬의 크기가 커짐에 따라 GPU 자원을 최대한 활용할 수 있어, 약 60 TFLOPS(tera floating point operations per second)에 가까운 성능을 달성하였다 (RTX 4090의 경우, FP32의 이론 최고 속도는 82.6 TFLOPS).

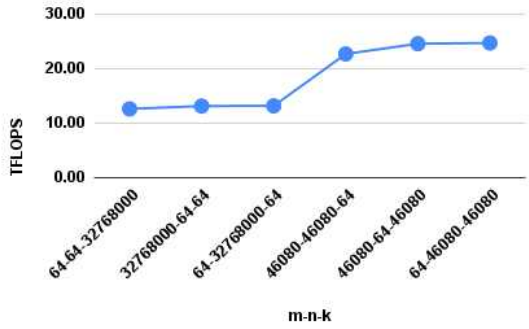


〔그림 2〕 정사각형 형태 행렬기반 행렬-행렬 곱셈 성능

Test Case	m	n	k	ops
1	64	64	32768k	268435456000
2	32768k	64	64	268435456000
3	64	32768k	64	268435456000
4	46080	46080	64	271790899200
5	46080	64	46080	271790899200
6	64	46080	46080	271790899200

〔표 1〕 좁은 사각형 형태 행렬기반 6가지 실험 사례

[표 1]은 [그림 1]에서 나타내는 좁은 사각형 형태들의 6가지 예제를 기반을 둔 실험 사례이다. 정사각형 형태 행렬기반 행렬-행렬 곱셈에서 달성한 약 60 TFLOPS 성능과 유사한 연산량을 기반으로 좁은 사각형 형태 행렬기반 사례들을 만들고자 했지만, 한정된 GPU 메모리 크기로 인해 약 40 TFLOPS 정도의 성능을 달성한 $m = n = k = 5120$의 연산량과 유사한 6가지 실험 사례를 보여준다. [그림 2]은 [표 1]에서 보여주는 6가지 실험 사례에 대한 실험 결과를 보인다. [그림 2]에서 볼 수 있듯이, m, n, k 중 하나의 크기만 큰 경우(1, 2, 3)가 약 13 TFLOPS로 약 24 TFLOPS를 달성한 두 개의 크기만 큰 경우(4, 5, 6)에 비해 낮은 성능을 보여준다. 그뿐만 아니라, 모든 6가지 경우 약 40 TFLOPS를 달성한 유사한 연산량을 갖는 정사각형 형태기반 행렬-행렬 곱셈에 비해 낮은 성능을 보여준다.



〔그림 3〕 좁은 사각형 형태 행렬기반 행렬-행렬 곱셈 성능

Ⅲ. 결론

본 논문에서는 좁은-사각형 형태의 행렬들을 기반한 다양한 비정형 행렬-행렬 연산에 대한 NVIDIA cuBLAS 서브루틴인 cublass<t>gemm() 함수의 성능을 분석하였다. 유사한 연산량임에도 불구하고 정사각형 형태의 행렬을 기반으로 둔 행렬-행렬 곱셈 연산의 성능이 좁은 사각형 형태의 행렬을 기반으로 두 행렬-행렬 곱셈 연산에 비해 높은 성능을 보여준다. 따라서, 대량의 데이터를 통해 행렬기반의 데이터를 구성할 때, 해당 행렬이 사용될 연산에 성능의 영향에 끼칠 수 있는 형태를 분석하고 이용하는 것이 중요하다.

ACKNOWLEDGMENT

이 논문은 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (P0020632, 2023년 산업혁신인재성장지원사업)

참 고 문 헌

[1] O'Leary, Daniel E. "Artificial intelligence and big data." IEEE intelligent systems 28, no. 2 (2013): 96-99.

[2] Rathore, M. Mazhar, Syed Attique Shah, Dhirendra Shukla, Elmahdi Bentafat, and Spiridon Bakiras. "The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities." IEEE Access 9 (2021): 32030-32052.

[3] Schmidt, Bertil, and Andreas Hildebrandt. "Next-generation sequencing: big data meets high performance computing." Drug discovery today 22, no. 4 (2017): 712-717.

[4] J. Kim, A. Panyala, B. Peng, K. Kowalski, P. Sadayappan and S. Kr- ishnamoorthy, "Scalable Heterogeneous Execution of a Coupled-Cluster Model with Perturbative Triples," SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, 2020, pp. 1-15.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[6] Subakan, Cem, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. "Attention is all you need in speech separation." In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 21-25. IEEE, 2021.

[7] NVIDIA cuBLAS Library, http://developer.nvidia.com/cublas.