CUDA 기반 숄레스키 분해 성능 최적화 환경 탐색

강준범 [1], 이명호 [2], 박능수 [1]
[1] 건국대학교 컴퓨터공학과
[2] 명지대학교 컴퓨터공학과

aopko@konkuk.ac.kr, myunghol@mju.ac.kr, neungsoo@konkuk.ac.kr

Exploration of Optimization Environment for CUDA-based Cholesky Decomposition

Junbeom Kang[1], Myungho Lee[2] , Neungsoo Park[1]
[1] Dept. of Computer Science and Engineering, Konkuk University
[2] Dept. of Computer Science and Engineering, Myongji University

요 약

최근 다양한 연구 분야에서는 CUDA 프레임워크를 이용하여 병렬 처리를 통해 연산 시간을 단축하는데 성공하고 있다. 이 중 숄레스키 분해는 양의 정부호 행렬을 하삼각행렬로 분해하는 과정에서 많은 행렬 곱셈이 요구되어 GPU 의 구조적 특징을 활용하면 상당한 가속화가 가능하다. 따라서 이 논문에서는 CUDA 코어에 연산을 할당할 때, 핵심 요소인 블록의 개수와 블록 당 쓰레드 개수를 조절할 수 있는 병렬 숄레스키 분해 연산 프로그램을 구현하였다. 서로 다른 세 종류의 행렬 크기에 대해 다양한 블록 수-쓰레드 수 환경을 설정하여 가속화 정도를 측정한 결과, 각 행렬 별 최적 환경에서 동일 그룹 내 최장 시간 대비, 1000x1000 행렬에서는 약 1.80 배, 2000x2000 행렬에서는 약 2.94 배의 추가적인 가속화를 달성하였다.

1. 서론

(RS-2023-00321688).최근 CUDA 프레임워크를 활용해 기존 순차 방식 프로그램에 병렬처리를 적용하여 그 성능을 향상시키는 연구가 다양한 분야에서 진행되고 있다[1][2]. 특히 행렬 연산이 많은 비중을 차지하는 인공지능 신경망 학습에서 CUDA 프레임워크를 활용해 가속화를 진행하는 연구가 진행되고 있다[3]. 위 연구 내용을 근거로, 숄레스키 분해 또한 행렬 연산을 주로 수행하기 때문에 CUDA 프레임워크를 활용하여 기존 순차 방식의 숄레스키 분해 프로그램을 병렬화 하여 연산 가속화가 가능하다고 판단하였다. 따라서 본 논문에서는 CUDA 기반 병렬 숄레스키 분해 연산 프로그램을 구현하고 GPU 디바이스 메모리 구조의 특징을 고려하여 행렬 크기마다 존재하는 최적의 블록 수와 블록 당 쓰레드 수를 조절하며 최적화 환경을 탐색하여 추가적인 가속화 방식을 제시한다.

2. 연구 배경

2.1 숄레스키 분해

$$A = LL^T$$

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & 0) \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \cdots (1)$$

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} \qquad \cdots (2)$$

$$l_{ij} = 1/l_{jj}(a_{ij} - \sum_{k=1}^{j-1} l_{ik} \cdot l_{jk}) \cdots (3)$$

숄레스키 분해는 양의 정부호인 행렬을 하삼각행렬과 그 전치행렬로 분해하는 연산으로, 수식 1 과 같은 과정으로 풀이된다. A 는 양의 정부호 행렬, L 은 하삼각행렬을 의미한다. 이 때, 각 원소들을 구하기 위해서 수식 (2)와 수식 (3)을 반복적으로 수행하여 원소를 하나씩 계산해낸다. 원소 간 의존성에 의해 병렬화가 어려웠으나 GPU 기반의 숄레스키 분해에 관한 연구가 지속되어 특별한 알고리즘을 통해 순차 알고리즘보다 더욱 뛰어난 가속화가 가능함을 증명했다[4].

2.2 CUDA(Compute Unified Device Architecture)

CUDA 는 NVIDIA 사에서 만든 GPU 를 사용하여 대규모 병렬처리 작업을 수행할 수 있도록 해주는 프레임워크이다[5]. 사용 방식은 GPU 디바이스 상에 물리적으로 존재하는 CUDA 코어를 연산 장치로 사용하며, 각 코어에 연산 작업을 할당하는 방식은 그림 1 과 같이 쓰레드 블록 구조로 구성되어 있다. Grid 는 동일한 쓰레드 개수를 가진 Block 의 그룹이며, Block 은 여러 개의 Thread 로 구성된 구조이다. 이 Thread 들은 한 개씩 CUDA 코어에 할당되어 작업을 수행하도록 한다. 따라서 수행하고자 하는 연산에 적절한 블록 수와 쓰레드 수를 가진 환경을 조성하면 메모리 액세스를 최적화할 수 있고, GPU 디바이스의 자원을 낭비 없이 사용할 수 있으며 작업부하 분산을 균일하게 분배할 수 있기 때문에 결과적으로 우수한 성능 향상과 프로그램 실행 시간 가속화를 기대할 수 있다.

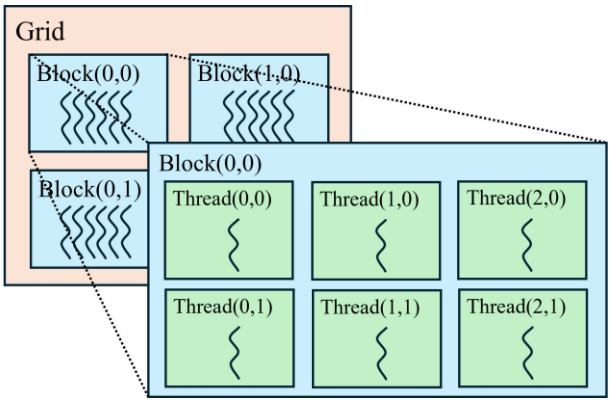

Figure 1 – Thread block architecture

2.3 CUDA 기반 병렬 숄레스키 분해 프로그램 구현

연구 배경을 토대로, 순차 알고리즘을 병렬처리를 위해 일부 변환하여 특정 원소를 계산하기 직전까지 구한 행렬 원소 정보를 바탕으로 동시에 다른 원소를 계산할 수 있도록 프로그램을 작성했으며, CUDA 에서 코어에 연산을 할당할 때 필수적인 Kernel 코드에 제공해줘야 하는 2 차원의 Grid 와 Block 을 변경하며 최적화 환경을 파악할 수 있도록 작성했다.

```
for i from 0 to n do
    for j from 0 to i do
        s = 0
        for k from 1 to j-1 do
            s = s + L[i,k] * L[j,k]
        end for
        if i == j then
            L[i,j] = sqrt(A[i,i] - s)
        else
            L[i,j] = (1.0 / L[j,j]) * (A[i,j] - s)
        end if
    end for
end for
```

Figure 2 – Serial Cholesky Decomposition pseudo-code

위 코드는 순차 알고리즘의 의사코드로, 삼중 반복문을 통해 원소 하나씩을 하삼각행렬의 크기만큼 L[0,0] 부터 시작하여 왼쪽 열부터 대각에 위치한 원소까지 계산하는 구조이다.

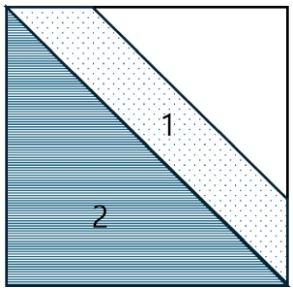

Figure 3 – Parallel Cholesky Decomposition

병렬처리로 구현한 숄레스키 분해 프로그램은 blockId 와 threadId 를 이용하여 우선적으로 대각 위치의 코어를 먼저 제곱근을 취하고 이전 원소들을 이용해 계산한다. 이 후, 대각 요소 기준으로 하삼각행렬의 각 요소를 나누는 작업을 수행한다. 대각에 위치한 원소 하단의 삼각행렬 원소들은 전체 행렬의 크기를 스레드의 수로 등분하여 처리할 작업의 범위를 기준으로 각자 담당한 원소 연산을 완료하여 병렬화를 극대화한다.

3. 실험

CUDA 기반의 병렬 숄레스키 분해 연산 프로그램의 최적 환경을 탐색하기 위해 사전 설정한 데이터는 표 1 과 같은 서로 다른 3 가지 크기의 양의 정부호 행렬을 만들고, 블록 개수와 블록 당 쓰레드 개수를 고정된 배열 내에서만 선택하여 사용하였다.

표 1. 사전 설정 데이터

행렬 크기(가로 x 세로)	1000x1000, 2000x2000
Block 개수	{16,64,256,1024}
Block 당 Thread 개수	{16,64,256,1024}

실험 환경은 표 2 와 같다. 시간 측정은 정밀한 측정을 위해 NVIDIA CUDA Toolkit 에서 제공하는 Nsight System 을 활용하여 kernel execution time 을 나노초(ns) 단위로 측정하였다.

표 2. 실험 환경

CPU	AMD Ryzen 7 7800x3D
GPU	NVIDIA RTX 4070ti
GPU CUDA Core	7,680 개
OS	Linux(WSL)

3.1 실험 방식

우선 C 언어를 이용한 순차 숄레스키 분해 프로그램의 실행 시간을 10 회 측정하여 그 평균을 순차 프로그램의 실행 시간으로 설정했다. 이 값과 비교하여 CUDA 기반의 병렬 숄레스키 분해 연산 프로그램을 블록 개수와 블록 당 쓰레드 수를 변경해가며 각 실행 시간을 측정하고 실행시간마다 발생한 가속화 수준을 계산했다.

3.2 실험 결과

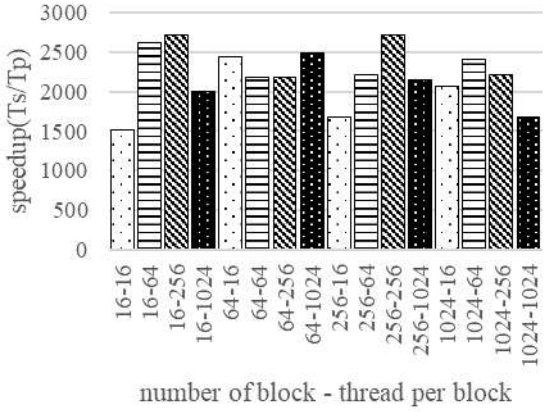

Figure 2 - 1000x1000 size matrix speedup

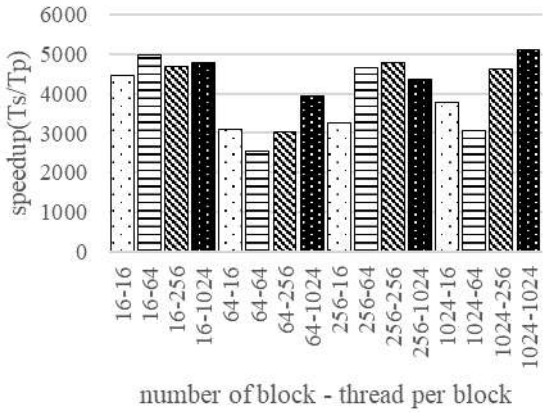

Figure 3 - 2000x2000 size matrix speedup

위 두 그림에 따라, 행렬의 크기에 따라 적절한 블록의 개수와 블록 별 쓰레드 개수가 존재함을 알 수 있다. 1000x1000 크기의 행렬에서는 블록 개수가 64 개일 때 평균적으로 약 2250 배의 다른 블록 수에 비해 가장 큰 속도 향상이 있었다. 마지막으로 2000x2000 크기의 행렬에서는 블록 개수가 16 개일 때 평균적으로 약 4670 배의 가장 큰 속도 향상이 있었다. 이를 통해 행렬 사이즈가 커질수록 설정한 블록의 개수가 작아졌을 때. 추가적인 가속화가 발생한 것을 알 수 있다. 행렬 크기 별 블록 간 최단 수행 시간과 최장 수행 시간을 비교하였을 때는 1000x1000 크기의 행렬에서는 1.80 배, 2000x2000 크기의 행렬에서는 2.94 배 만큼 연산 수행 시간이 빨라졌다.

4. 결론

본 연구는 블록 수와 블록 당 쓰레드 수를 조절가능한 CUDA 기반의 병렬 숄레스키 분해 연산 프로그램을 구현하여 순차 숄레스키 분해 연산 프로그램의 수행 시간과 비교하였을 때 뛰어난 가속화 수준을 보이는 최적 환경을 탐색하기 위해 설계되었다. 실험을 통해 숄레스키 분해 연산을 수행하고자 하는 행렬의 크기가 증가할수록 블록 개수를 작은 값에서 설정하면 더욱 우수한 가속화를 기대할 수 있음을 발견했다. 추후 추가적인 연구를 통해 디바이스 의존적인 현재 프로그램을 개선하여 숄레스키 분해를 위한 블록 수 최적 환경 유도 공식을 만들고 디바이스에 독립적으로 적용할 수 있는 프로그램을 연구하고자 한다.

감사의 글

본 연구는 과학기술정보통신부의 재원으로 한국연구재단의 지원 사업(RS-2023-00321688)과 정보통신기획평가원의 정보통신방송혁신인재양성(메타버스융합대학원)사업(IITP-2024-RS-2023-00256615)의 연구 결과로 수행되었음

참고문헌

[1] 김호중, 조태훈, "GPU 를 이용한 위상 측정법의 가속화," 한국정보통신학회논문지, Vol.21, No.12, pp.2285-2290, 2017.

[2] 서지완, 박채림, 조세홍, 계희원, "의료영상을 위한 위치 기반 역학의 GPU 병렬화 연구," 한국차세대컴퓨팅학회 논문지, Vol.19, No.3, pp.19-28, 2023.

[3] Salles Civitarese, Daniel & Szwarcman, Dilza & Vellasco, Marley. Speeding Up the Training of Neural Networks with CUDA Technology. Zakopane, Poland. 2012. pp.30-38.

[4] Azzam Haidar, Ahmad Abdelfatah, Stanimire Tomov, and Jack Dongarra. High-performance Cholesky factorization for GPU-only execution. In Proceedings of the General Purpose GPUs (GPGPU-10). New York, NY, USA, 2017. pp.42–52.

[5] NVIDIA, CUDA C++ Programming Guide, https://docs.nvidia.com/cuda/cuda-c-programming-guide/