병렬 프로그래밍 중간 보고서

팀명: TEAM_C_TENCORE

팀명: 소프트웨어 융합대학 20143148 이호중

소프트웨어 융합대학 20135357 한승탁 컴퓨터공학과 20135114 김용화 콘텐츠 IT 20125345 조준형

목표: Tensor core 기반의 행렬곱 가속화 및 딥러닝 적용 방식 분석 기간:2018 . 09 .01 ~ 2018. 12.20

1.일정

1.20	
기간	주요내용
2018.09.01~ 2018.10.29	To implement cpu
2018.10.30~ 2018.11.06	To setup the CUDA development environment To implement general GPU version
2018.11.06~ 2018.11.13	To learn the Tensor Core document
2018.11.14~ 2018.11.26	To implement Tensor Core version
2018.11.27~ 2018.12.03	Experiments and analysis
2018.12.04~ 2018.12.07	To write a report

2.진행사항

미팅 날짜	내용
10월 9일	-Tensor Core 기반의 행렬곱 가속화 및 딥러닝 적용 방식 분석으로 주제를 재확인 하였다.
	-다음 미팅 일정은 중간고사를 고려하여 10월 29일로 한다. -10월 29일 까지 다음 과제를 완성해야한다. 1.행렬의 연산 법칙 확인
	2-1.C/C++ 언어로 CPU상에서 행렬의 연산 함수를 만든다. 2-2.작성한 행렬의 연산 함수들을 병렬화 시키는 방법을 글로 작성한다. 3-1.추가 사항 멀티 쓰레드르 이용 2번에서 작성한 함수를 병렬화 해본다.
	3-2.행렬의 연산을 간력화시킨 알고리즘들을 알아보고 병렬화가 가능한지 알아본다. - 첨부 2-1 과제 pdf파일 - 첨부 2-2 발표 ppt파일

미팅 날짜	내용
10월 29일	10월 9일 과제 발표 - CPU를 사용한 연산은 CPU 코어 수 이하로 쓰레드를 생성하도록 변경할 것 * CPU 코어수를 초과한 쓰레드는 스케쥴링이 되어서 시간이 더 걸리게 된다. * 추가적으로 AXB 이후 C연산시 상호배제, atom 연산 사용할 것 10월 29일 과제 - CUDA의 CUBLAS 라이브러리를 이용 해당 행렬 A*B+C 연산을 작성할 것 첨부 2-3 미팅 모임 사진

첨부 2-1 10월 9일 대학생들 과제 pdf

10.29 일까지 다음과 같이 내용을 완성 바랍니다.

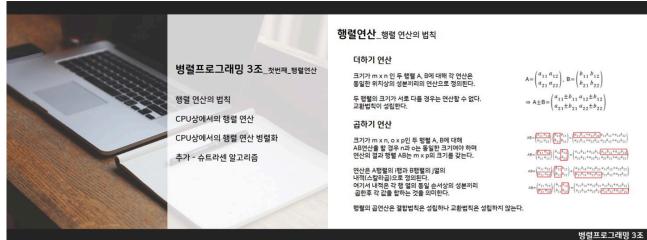
- 1. Matrix 의 더하기와 곱하기 연산의 법칙을 복습하세요. (글로 정리)
- 2. C/C++이용하여 CPU 상에 Matrix 의 더함과 곱함 함수를 만들어보세요.
 - * 예) matrix_add()로 A+B 계산함, matrix_multiplication()는 AxB 계산함.
 - matrix 의 shape 가 parameter 로 지정할수 있다.
 - 아래 행렬을 계산하고 출력해보세요.

If
$$A = \begin{bmatrix} 1 & 0 & -3 \\ -2 & 4 & 1 \end{bmatrix}$$
 and $B = \begin{bmatrix} 2 & -1 \\ 3 & 0 \\ -5 & 2 \end{bmatrix}$ and $C = \begin{bmatrix} 3 & -1 \\ -2 & 2 \end{bmatrix}$
Then

- $A \times B = ? \text{ and } A \times B + C = ?$
- * 작성한 더함과 곱함 함수들이 병렬화 시키는 방법을 생각해보세요.
 - 나중에 보고서를 쓰실때 도움이 될거니까 글로 정리하세요
- 3. 추가 사항 (Optional):
 - * Multi-thread 로 위의 matrix_add(), matrix_multiplication()함수 구현해보세요. (win32
- api 나 pthread 등 library 이용)
 - * 수학자들이 간화시킨 알고리즘들 알아보고 병렬화가 가능한지 생각해보세요.

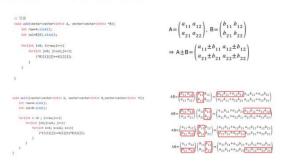
위의 내용을 정리하고 발표자료 (연산의 법칙, 구현한 코드, 실행결과 캡쳡, 병렬화의 생각, 어려운점등 포함) 만들어서 발표하세요.

첨부 2-2 10월 9일 과제 발표 ppt

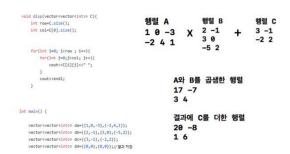


워틸프포그네당 25

행렬연산_CPU상에서의 행렬 연산



행렬연산_CPU상에서의 행렬 연산



병렬프로그래밍 3조 병렬프로그래밍 3조

<mark>행렬연산_CPU</mark>상에서의 행렬 연산 병렬화



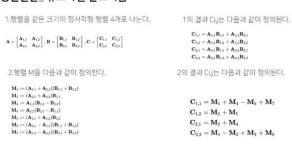
A. 행렬 연산을 하는 함수에 계산할 행의 정보를 인자로 받을 수 있도록 만들었다.

행렬연산_CPU상에서의 행렬 연산 병렬화



병렬프로그래밍 3조 병렬프로그래밍 3조

행렬연산_ 슈트라센 알고리즘



ㅁ행렬의 전체 곱셈을 2번 과정처럼 7번의 곱셈과 18번의 덧셈으로 처리할 수 있다.

 $\rightarrow 7 \cdot n^{\log_2 7} - 6 \cdot n^2$

행렬연산_ 슈트라센 알고리즘

알고리즘	시간 복잡도
00 antidensemble is administration betweentien (Q) for execution for exe	O(n³)
$\begin{array}{lll} \Phi(M_1,d_1) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\ & > \Phi(M_1,d_2) & > \Phi(M_1,d_2) \\$	O(n ^{2.807})

병렬프로그래밍 3조 병렬프로그래밍 3조

행렬연산_ 슈트라센 알고리즘_병렬화

$$\begin{split} \mathbf{A} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{13} & A_{23} \end{bmatrix}, \mathbf{B} &= \begin{bmatrix} B_{11} & B_{12} \\ B_{13} & B_{23} \end{bmatrix}, \mathbf{c} &= \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \end{split} \qquad \quad \mathbf{A}. \vec{\mathbf{w}} \vec{\mathbf{g}} \\ \mathbf{B} &= \begin{bmatrix} \mathbf{A}_{11} + \mathbf{A}_{22} \end{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{12} &= (\mathbf{A}_{11} + \mathbf{A}_{22}) \mathbf{B}_{11} \\ \mathbf{B}_{12} &= \mathbf{A}_{13} (\mathbf{B}_{12} - \mathbf{B}_{12}) \\ \mathbf{M}_{13} &= \mathbf{A}_{13} (\mathbf{B}_{12} - \mathbf{B}_{13}) \\ \mathbf{M}_{13} &= (\mathbf{A}_{13} + \mathbf{A}_{12}) \mathbf{B}_{22} \\ \mathbf{M}_{14} &= (\mathbf{A}_{13} - \mathbf{A}_{13}) (\mathbf{B}_{13} + \mathbf{B}_{12}) \\ \mathbf{M}_{15} &= (\mathbf{A}_{12} - \mathbf{A}_{22}) (\mathbf{B}_{23} + \mathbf{B}_{22}) \end{split} \qquad \qquad \qquad \quad \mathbf{B}. \mathbf{Y} \mathbf{Y} \mathbf{O} \quad \mathbf{M} \vec{\mathbf{i}} \vec{\mathbf{e}} \quad \vec{\mathbf{M}} \vec{\mathbf{i}} \vec{\mathbf{e}} \vec{\mathbf{i}} \vec{\mathbf{e}} \vec{\mathbf{e$$

병렬프로그래밍 3조

https://github.com/HoJungLee/Parallel_Programming_2018_Fall/blob/master/Capstone/TEAM_C_TENCORE/matrix.cpp

첨부 2-3 10월 29일 미팅 사진

