

COMP1680 (2023/24)	클라우드, 그리드 및 가상화		기부금: 물론 60%
코스 리더: 캐서린 톨리 박사	교과 과정 2: MP 열기		마감일: 2023년 12월 13일 수요일
이 교과 과정은 최신 튜토리얼 작업을 숙지한 일반 학생이 수강해야 합니다. 약 30시간 피드백과 성적은 일반적으로 수업 마감일로부터 영업일 기준 15일 이내에 제공됩니다.			
학습 결과: 고성능 컴퓨팅 기반 아키텍처와 해당 애플리케이션에 대한 적합성을 특성화하고 비판적으로 평가합니다. 공유 및 분산 메모리 프로그래밍 패러다임을 사용하여 애플리케이션을 구현하고 실행합니다. 클라우드 및 그리드 컴퓨팅의 역할과 애플리케이션을 설명하고 비판적으로 논의합니다.			

표절은 다른 사람의 작업을 자신의 작업인 것처럼 제시하는 것입니다. 여기에는 자료를 참조하지 않고 웹이나 책에서 직접 정보를 복사하는 행위가 포함됩니다. 개별 노력으로 공동 교과 과정을 제출합니다. 다른 학생의 수업 내용을 복사하는 것; 다른 학생의 수업 과제를 훔쳐서 자신의 과제로 제출하는 행위. 표절이 의심되는 경우 조사를 실시하며, 표절이 확인된 경우에는 대학이 정한 절차에 따라 처리됩니다. 무엇이 표절인지 아닌지에 대한 자세한 내용은 학생 핸드북을 참조하세요.

모든 출처(예: 인터넷, 서적)에서 복사되거나 수정된 모든 자료는 사용 중인 참조 스타일에 따라 올바르게 참조되어야 합니다.

귀하의 작업은 표절 검사를 위해 제출됩니다. 표절 탐지 시스템을 우회하려는 모든 시도는 심각한 평가 위반으로 처리됩니다.

교과 과정 제출 요구 사항

- 이 교과 과정에 대한 전자 사본은 COMP1680 교과 과정 Moodle 페이지의 링크를 사용하여 마감일인 2022년 13월 12일 수요일 23:30까지 완전히 업로드되어야 합니다.
- 이 교과 과정을 위해서는 PDF 문서와 코드가 포함된 zip 파일을 제출해야 합니다. 일반적으로 문서의 텍스트는 이미지가 아니어야 하며(예: 스캔해서는 안 됨) 일반적으로 다른 문서(예: "다른 이름으로 저장 .. PDF"를 사용하는 MS Office)에서 생성됩니다. 이에 대한 예외는 손으로 쓴 수학 표기법이지만 스캔할 때 파일 크기가 과도하지 않은지 확인하십시오.
- 파일 크기에 제한이 있습니다(해당 강좌 Moodle 페이지 참조).
- 업로드하는 모든 파일에는 바이러스가 없고 비밀번호로 보호되거나 손상되지 않았는지 확인하십시오. 그렇지 않으면 해당 파일은 null 제출로 처리됩니다.
- 이 교과 과정의 종이 사본을 제출해서는 안 됩니다.
- 모든 교과목은 위와 같이 제출되어야 합니다. 어떤 경우에도 이메일이나 기타 수단을 통해 교직원이 수락할 수 없습니다.

대학 웹사이트에는 늦은 제출에 대한 처벌 세부 사항, 참작할 수 있는 상황에 대한 절차, 평가 위반에 대한 처벌을 포함하여 현재 교과 과정 규정에 대한 세부 정보가 있습니다. <http://www2.gre.ac.uk/current-students/regs>를 참조하세요.

상세사항

이 과정은 개별적으로 완료해야 합니다.

이 과제를 완료하려면 다음 URL에 제공된 소스 코드가 필요합니다.

<https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601163>

<https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601162>

Jacobi 및 Gauss-Seidel 반복 방법을 사용하여 직사각형 2차원 열전도 문제를 해결하는 두 개의 C 프로그램 코드 (jacobi2d.c 및 gauss2d.c)가 제공됩니다.

이 코드는 컴파일 및 링크되어 jacobiSerial이라는 기존 실행 파일을 생성할 수 있습니다.

다음 명령을 사용하여 gaussSerial을 실행합니다.

```
gcc jacobi2d.c -o jacobi2d.out
```

```
gcc gauss2d.c -o gauss2d.out
```

실행 파일을 실행하려면 제공된 slurm 스크립트를 사용하세요.

<https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601167>

<https://moodlecurrent.gre.ac.uk/mod/resource/view.php?id=2601165>

다음 4단계를 각각 구현할 때 이전 버전의 솔루션을 유지하고 덮어쓰지 않도록 하십시오. 각 단계마다 하나씩 각 코드의 4가지 버전을 제출해야 합니다.

University HPC를 사용하여 코드를 컴파일하고 실행합니다. 이는 대기열이 있는 공유 리소스이므로 날짜가 가까워지면 바빠질 수 있으므로 코드를 실행하는 데 충분한 시간을 주고 마지막 순간까지 기다리지 않도록 하십시오. HPC 사용 방법을 잘 모르는 경우 실험실 노트와 Moodle의 지침을 확인하세요.

이러한 작업을 주의 깊게 따르고 제공된 코드를 사용하면 다른 Jacobi 및 Gauss 코드를 기반으로 한 작업은 표시되지 않으며 1점을 받고 더 이상 표시되지 않습니다.

잘못된 병렬화를 최적화하면 코드가 손상될 수 있으므로 모든 단계에서 여전히 동일한 대답을 제공하는지 확인하세요.

보고서는 2000단어로 작성되어야 하며 단지 코드와 결과의 스크린샷이 되어서는 안 됩니다.

당신이 한 일과 그 이유에 대해 논의해야 합니다. 또한 코드는 보고서에만 표시되지 않으므로 보고서에 코드 변경 사항을 문서화해야 합니다.

필요한 모든 작업을 완료하세요.

과제 1(25점)

다양한 문제 크기와 함께 상단 10°C, 하단 30°C, 왼쪽 40°C, 오른쪽 50°C로 경계 조건이 설정된 플레이트를 시뮬레이션하는 직사각형 2D 헤드 전도 문제에 대한 온도 분포를 계산해야 합니다. 이렇게 하려면 코드를 다음과 같이 수정해야 합니다.

1. 위에서 설명한 경계 조건을 반영합니다. 2. 실행 시간을 보고합니다.

문제 크기 범위에 따라 코드의 런타임을 기록합니다.

다양한 수준의 컴파일러 최적화 사용(예: -O1, -O2 등)

보고서에 코드 변경 사항을 문서화하세요. 코드를 제출하더라도 별도로 점수를 받을 수 없으므로 보고서에서 변경 사항을 강조하는 것이 중요합니다. 변경 사항을 설명하세요.

과제 2(30점)

그런 다음 OpenMP를 사용하여 코드의 기본 병렬 버전을 생성하려면 1단계에서 만든 응용 프로그램을 수정해야 합니다. 다음 명령은 OpenMP가 설치된 플랫폼에서 병렬 버전을 컴파일합니다.

```
gcc -fopenmp jacobiOpenmp.c -o jacobiOpenmp
```

```
gcc -fopenmp gaussOpenmp.c -o gaussOpenmp
```

병렬 코드에는 코드의 병렬 런타임을 보고하는 타이머가 포함되어야 합니다. 이 버전은 성능에 관계없이 1, 2, 4, 8 및 16개의 스레드를 사용하여 올바른 작동을 확립하기 위해 테스트되어야 합니다.

보고서에 1,2,4, 8, 16의 20x20 문제 크기에 대한 온도 인쇄 내용을 포함시키세요.
코드가 올바르게 작동하는지 보여주는 스레드입니다.

보고서에 코드 변경 사항을 문서화하세요. 코드를 제출하더라도 점수를 받을 수 없으므로 보고서에서 변경 사항을 강조하는 것이 중요합니다. 변경된 사항을 설명하세요.

20x20 문제 크기에 대해 1개 및 2개의 스레드를 사용하여 1번만 반복하여 Gauss-Seidel 코드를 실행합니다.
타이밍과 함께 온도를 출력하고 이를 보고서에 포함합니다. 해결책의 차이에 대한 이유를 토론하십시오.

과제 3(30점)

대학 HPC를 사용하여 2단계에서 생성한 OpenMP 구현으로 성능 테스트를 실행합니다. 이를 위해서는 코드에서 대부분의 인쇄 출력을 제거하고 문제 크기를 늘려 유용한 속도 향상을 입증할 수 있는 충분한 작업을 제공해야 합니다. 속도 향상 결과를 제공할 것으로 예상됩니다.

1. 세 가지 이상의 문제 크기에 대해 작은 도메인의 경우 속도 향상이 거의 보이지 않으며 최소 100x100 그리드와 일관된 허용 오차(최대 10-3)를 사용합니다.
2. 스레드 수 범위(2~8개 스레드) 병렬 코드의 속도 향상을 계산할 때 1단계에서 생성한 코드의 최적화된 단일 프로세서 버전을 사용하고 이를 비교해야 합니다. 유사한 컴파일러를 적용해야 합니다.

병렬 코드 최적화. 적절한 단위로 런타임을 나열하세요.

시리얼 버전에서 투어 타이밍과 속도 향상을 모두 보고해 주세요. 속도 향상에 대해 설명하고 이론상의 최대치와 비교하면 어떻습니까?

과제 4(15점)

다른 OpenMP 지시어와 절을 사용하면 OpenMP 애플리케이션을 추가로 수정하여 병렬 성능을 향상시킬 수 있습니다. 3단계에서 얻은 결과와 비교할 수 있는 결과를 제공해야 합니다. Jacobi 방법과 Gauss-Seidel 방법 최적화 간의 차이점에 대해 설명하십시오. 코드에 대한 변경 사항을 문서화하고 해당 변경 사항을 적용한 이유를 설명하십시오.

결과물

• 보고서가 포함된 PDF 파일

• 솔루션의 소스 코드가 포함된 ZIP 파일.

귀하의 보고서는 위에서 설명한 대로 1~4단계 구현에 대한 세부정보를 제공해야 합니다.
보고서에는 귀하의 솔루션에 대한 논의가 포함되어야 하며 다음에 대한 명확한 설명이 제공되어야 합니다. 구현한 코드 변경 사항, 컴파일 및 실행 프로세스, 테스트 사례. 3단계와 4단계에서는 표 및 그래픽 결과를 제공해야 합니다. 두 방법의 차이점과 병렬화에 미치는 영향에 대해 설명합니다.
zip 파일은 각 구현에 대해 적절하게 이름이 지정된 소스 코드 파일을 제공해야 합니다. 보고서는 코드 조각과 결과 표를 제외하고 약 2000단어로 구성되어야 합니다.

등급 기준

귀하는 코드 조각, 타이밍 및 결과에 대한 토론을 기반으로 점수를 받게 됩니다. 보고서는 올바른 영어로 작성되어야 하며, 명확하지 않은 내용으로 인해 감점될 수 있습니다.

다음 기준표에 따라 할당된 점수:

	1 단계 25점	2 단계 30점	3단계 30점	4단계 15점
1단계: 0-12점 2,3단계: 0-14점 4단계: 0-7점	안 불만족스러운 단계, 코드 타이밍 또는 경계 조건 변경이 부족합 니다.	안 불만족스럽다 단계, 코드에 필요한 병 렬화 단계가 부족합니다.	안 불만족스럽다 단계에서 코드 타이 밍에 대한 증거가 없거나 매우 제한적입니다.	불만족스러운 단계, 증거가 거의 없습니 다 병렬 코드를 최 적화합니다.
1단계: 13-14 마르크 2,3단계: 16-17점 4단계: 8점	만족스러운 단계이 지만 오류가 있습니다. 코드에 오류가 있거 나 보고서에 세부 정보 가 부족합니다.	만족스러운 단계이 지만 오류가 있습니다. 코드에 오류가 있거 나 보고서에 세부 정보 가 부족합니다.	그러나 만족할 만한 단계는 보 고서의 타 이밍에 오류가 있거나 세부 정보가 부족하 다는 것입니다.	만족스러운 단계이지만 오류 가 있습니다. 최적화 또는 세부 정보 부족 보고서
1단계: 15-17 마르크 2,3단계: 18-20점 4단계: 9-10점	좋은 단계이지만 코드는 대부분 정확 하고 문서화 하지만 거기에는 부족 토론 결과.	좋은 단계이지만 코드는 대부분 정확 하고 문서화 하지만 거기에는 부족 토론 결과.	좋은 조치이지만 타이밍은 대부분 정 확하고 문서화 하지만 거기에는 부족 결과에 대한 토론.	그러나 좋은 단계 는 최적화가 대부분 정확하고 문서화되어 있지만 부족하다 에 대한 토론 결과.
1단계: 18-19 마르크 2,3단계: 21-23점 4단계: 11점	아주 좋은 그러나 단계에서 는 코드와 타이 밍이 정확하 지만	아주 좋은 그러나 코드는 다음 과 같습니다. 맞지만 몇 가지가 있습니다	아주 좋은 그러나 단계는 타이밍이 정 확하지만	아주 좋은 단계이지만 최 적화는 정확하지만 몇 개야

	보고서에는 몇 가지 제한 사항이 있습니다.	보고서의 한계.	몇 가지가 있습니다 제한 사항 보고서.	보고서의 한계.
1단계: 0-12점 2,3단계: 24-26점 4단계: 12-13 마크	우수한 그러나 단계에서는 코드와 타이밍은 정확하지만 부족하다 정말 중요한 보고서 분석	우수한 그러나 코드는 다음과 같습니다. 맞지만 부족하다 보고서의 진정한 비판적 분석	우수한 , 타이밍은 정확하지만 만 부족함이 있다 보고서의 진정한 비 판적 분석	훌륭한 단계이지만 최적화 는 정확하지만 사실이 부족하다 보고서의 비판적 분석.
1단계: 0-12점 2,3단계: 27-28점 4단계: 14-15점	뛰어난 단계이며 코드 와 타이밍이 정확하 며 보고서에 자세히 설 명되어 있습니다.	그러나 눈에 띄는 단 계는 코드가 정확하 고 보고서에 코드가 완전히 자세 히 설명되어 있다 는 것입니다.	안 미해결이지만 타이밍이 정확 하고 보고 서에 자세히 설명되어 있 습니다. 충분히.	그러나 뛰어난 단 계는 최적화가 정확했 으며 보고서에 이 에 대한 자세한 내 용이 나와 있다 는 것입니다.

이러한 지침 중 확실하지 않은 것이 있으면 모듈 리더에게 이메일을 보내거나 가능한 한 빨리 만나보실 수 있도록 약속을 잡으십시오.