

## 문제 1 (학부/대학원 공통): 유한차분 계산 병렬화

(배점: 학부 30 점 / 대학원 20 점)

### 문제 설명

본 문제는 주어진 유한차분 계산 순차 프로그램을 병렬화하는 것이 목적이다. 유한 차분 계산의 고유 특성상 계산 노드들 사이의 최소한의 정보 통신이 불가피하게 수반되며, 이를 고려하여 주어진 순차 프로그램을 병렬화 한다. 단, 아래의 조건들을 모두 만족시켜야 한다. (FORTRAN 또는 C 둘 중 하나의 언어를 선택해서 병렬화한다.)

(1) 주어진 순차 프로그램에서 이미 선언된 배열들은 전체 배열 형태로 모든 계산 노드가 공유하도록 한다. 순차 프로그램에서 제공된 배열, 변수들 이름을 그대로 유지한다. 병렬화를 위하여 배열 및 변수들을 추가적으로 선언할 수 있다.

(2) 주어진 순차 프로그램에서 수행하는 계산들은 여러 계산 노드에서 나누어서 계산할 수 있다. 특별히 변화시키지 말아야 할 표시된 부분은 그대로 둔다. 병렬화 전후에 동일한 계산을 할 수 있어야 한다.

(3) 일반적인 유한차분 계산의 특성을 고려한다. 데이터의 통신이 반드시 필요한 부분을 숙지하고, 계산 노드들 사이의 데이터 통신이 반드시 필요한 변수 및 배열을 찾아내고 효율적인 통신을 수행한다.

(4) 최종 계산 값을 계산 노드 숫자에 따라서 확인한다. 수치 계산적으로 동일한 병렬 계산 결과를 얻을 수 있어야 한다.

(5) 병렬 효율성을 확인한다. 계산 노드 수가 증가 할 때, 계산 노드 수가 한 개일 때 대비, 병렬 계산이 얼마나 빨라지는지를 체크한다.

(6) 순차 프로그램 계산에서 호출되는 `genw` 함수는 변화시키지 않는다. (해당 함수를 최적화 및 병렬화하지 않는다.) 주어진 순차 프로그램들의 주 프로그램에서만 병렬화 작업을 수행한다.

## 평가 방법

- (1) 학부 32 CPU, 대학원 64 CPU 를 이용해서 병렬화된 코드의 실행시간을 `time command` 로 측정. 모범답안으로 제시될 Reference 병렬 코드 대비 실행시간의 비율로 순위 결정.
- (2) 실행 결과값 (tmp) 이 CPU 개수와 상관없이 동일하게 나타나는지의 여부로 코드의 정확도를 측정. 결과값이 Round-off 에러를 벗어난 경우 정확성 검증 실패로 결격처리.