

[2014 제2회 PNU 병렬컴퓨팅 경진대회 문제]

□ 프로그래밍 환경

- 프로그래밍 언어 : C 또는 Fortran
- 컴파일러 : gcc
- 병렬화 라이브러리 : OpenMP, MPI

□ 제출 자료

- Makefile 또는 컴파일 환경(완전한 커맨드라인 포함)
- 실행파일과 소스파일

□ 제출 방법

- 이메일 제출 : choiseok@pusan.ac.kr
- 상기 제출자료를 zip으로 압축하여, "2014_PNU_경진대회_팀명.zip"으로 제출

□ 평가방법

- 결과 값의 품질 및 프로그램 수행속도를 통한 정량적 평가
 - 수행속도(70%), 결과 값의 품질(30%) : 병렬컴퓨팅 경진대회 취지 고려
- 정량적 평가로 분별이 힘든 순위의 해당 팀이 있을 경우 문제의 이해도, 코딩의 수준, 창의력에 기반한 정성적 평가

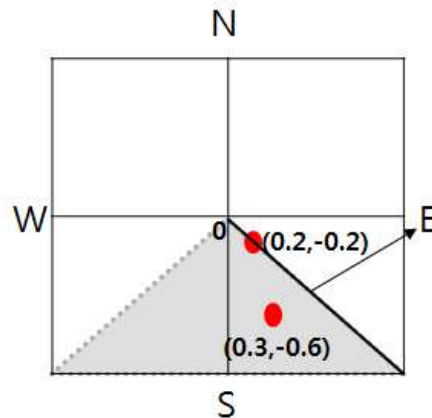
시험 1 (시뮬레이션)

문제 1.

한 개의 입자(particle)가 2차원 정사각형 격자(square grid) 상에서 다음과 같은 무작위 운동(random walk)을 한다. 즉, 입자는 매번 4개의 최근접 이웃(nearest neighbor) 격자점들 중 하나로(즉, 현 위치 기준으로 동,서,남,북에 위치한 지점으로) 이동한다. 만일, 입자가 이러한 무작위 이동을 N 번 수행한 경우에 입자가 초기 위치로부터 이동한 거리와 평균 이동거리를 계산하는 병렬코드를 만들어라. (코어는 최대 8개까지 사용) 단, 아래의 3가지 조건을 만족해야 한다.

1. 무작위 운동은 random하게 정해진 2차원 점 (X, Y) 에 의해 결정되고, X, Y 의 범위는 각각 $[-1,1]$ 이다. 아래 그림과 같이 (X, Y) 위치를 기준으로 가장 가까운 지점(동,서,남,북)에 해당되는 지점의 방위로 이동한다. (경계 조건에서는 반시계 방향으로 움직인다.)

ex) (X,Y) 의 값이 $(0.3, -0.6)$ 일 경우, 입자의 무작위 운동방향은 남쪽이다.
 (X,Y) 의 값이 $(0.2, -0.2)$ 일 경우, 입자의 무작위 운동방향은 동쪽이다.



2. $2N$ 개의 난수는 단일 코어를 사용하여 생성한다.
 - (X,Y) 의 쌍으로 N 개를 생성. 즉 발생된 난수가 $r_1, r_2, r_3, r_4, \dots, r_{2n-1}, r_{2n}$ 과 같을 때, $(r_1, r_2), (r_3, r_4), \dots, (r_{2n-1}, r_{2n})$ 가 됨
 - N 은 2^{30} 이다.
3. 난수의 seed는 20140827의 상수 값으로 고정한다.

시험 2 (2D Matrix Multiply)

주어진 실수 2차원 행렬 A와 B에 대한 행렬의 곱인 $AB=C$ 에서 C를 구하는 병렬 코드를 작성하시오. 단, A와 B 행렬의 크기 및 값은 아래와 같다.

A 행렬의 크기 : 2000×1500

A 행렬 값 : $A_{ij} = \frac{i+1}{j+1}$

B 행렬의 크기 : 1500×2000

B 행렬 값 : $B_{ij} = \frac{j+1}{i+1}$

문제 2.

OPENMP를 이용한 병렬 프로그래밍 코드를 작성하시오.

문제 3.

MPI를 이용한 병렬 프로그래밍 코드를 작성하시오.
(최대 주어진 코어 수는 $16[2\text{node} * 8\text{core}]$)

Serial Code - C language

```
#include <stdio.h>
#include <omp.h>

#define M 2000 // the number of rows in matrix A
#define N 1500 // the number of column in matrix A(rows in matrix B)
#define O 2000 // the number of column in matrix B

int main(int argc, char **argv)
{
    int i, j, k;
    double a[M][N],b[N][O],c[M][O];

    // Initialize Matrices
    for(i=0 ; i < M ; i++)
        for(j=0 ; j < N ; j++)
            a[i][j]= (double)(i+1)/(double)(j+1);
    for(i=0 ; i < N ; i++)
        for(j=0 ; j < O ; j++)
            b[i][j]= (double)(j+1)/(double)(i+1);
    for(i=0 ; i < M ; i++)
        for(j=0 ; j < O ; j++)
            c[i][j] = 0.0;

    // Calculate Matrix Multiply
    for(i=0 ; i < M ; i++){
        for(j=0 ; j < O ; j++)
            for(k=0 ; k < N ; k++)
                c[i][j] += a[i][k]*b[k][j];
    }
    return 0;
}
```

Serial Code - FORTRAN90 language

```
program matrix_multiply
  implicit none

  integer, parameter :: M=2000,N=1500,O=2000
  integer :: i,j,k
  real*8, dimension(M,N) :: a
  real*8, dimension(N,O) :: b
  real*8, dimension(M,O) :: c

  c=0.0
  do j = 1, N
    do i= 1, M
      a(i,j)=real(i)/real(j)
    enddo
  enddo

  do j = 1, O
    do i= 1, N
      b(i,j)=real(j)/real(i)
    enddo
  enddo

  do k=1,N
    do j=1,O
      do i=1,M
        c(i,j) = c(i,j)+a(i,k)*b(k,j)
      enddo
    enddo
  enddo

end program
```