

Homework Assignment 7 – due on Saturday, November 18 (Midnight)

Description of Assignment:

Complete an MPI program(matadd.c) that

- (i) decomposes A and B on p_0 into local_A and local_B on all processors
- (ii) computes local_C = local_A to local_B
- (iii) composes local_C on all processors into C on p_0 .

<pre>#include <stdio.h> #include <stdlib.h> #include <math.h> #include "mpi.h" #define N 12 float **malloc_2d(int row, int col) { float **A, *ptr; int len, i; len = sizeof(float *)*row + sizeof(float)*col*row; A = (float **)malloc(len); ptr = (float *) (A + row); for(i = 0; i < row; i++) A[i] = (ptr + col*i); return A; } main(int argc, char* argv[]) { float A[N][N], B[N][N], C[N][N]; float **local_A, **local_B, **local_C; int local_N, *displs, *counts, i, j, n; int np2, np, pid; MPI_Init(&argc, &argv); MPI_Comm_rank(MPI_COMM_WORLD, &pid); MPI_Comm_size(MPI_COMM_WORLD, &np2); np = sqrt(np2); local_N = N/np; local_A = malloc_2d(local_N, local_N); local_B = malloc_2d(local_N, local_N); local_C = malloc_2d(local_N, local_N);</pre>	<pre>// initialization of A and B if (pid == 0) { for (i=0; i<N; i++) for (j=0; j<N; j++) { A[i][j] = i*N+j; B[i][j] = N*N-i*N-j; } } // (i) decompose A and B into local_A and local_B displs = (int*)malloc(sizeof(int)*np2); counts = (int*)malloc(sizeof(int)*np2); // (ii) local_C = local_A + local_B // (iii) compose local_C to C // check the results if (pid == 0) for (i=0; i<N; i++) { for (j=0; j<N; j++) printf("%3.0f ", C[i][j]); printf("\n"); } free(local_A); free(local_B); free(local_C); free(displs); free(counts); MPI_Finalize(); }</pre>
--	--

How to proceed:

- (i) Use MPI_Scatterv and MPI_Gatherv in your program.
- (ii) Run only 4(=2x2), 9(=3x3) or 16(=4x4) processors for tests. The program prints same results regardless the number of processes.

Turnin the assignment:

After done your assignment, type **turnin** in your current working directory. You can retype the command at any time before the due date.