НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №8
з дисципліни «Програмування паралельних комп'ютерних систем»

Виконала:
студентка 3 курсу
ФІОТ гр. ІО-21
Сурай О. В.

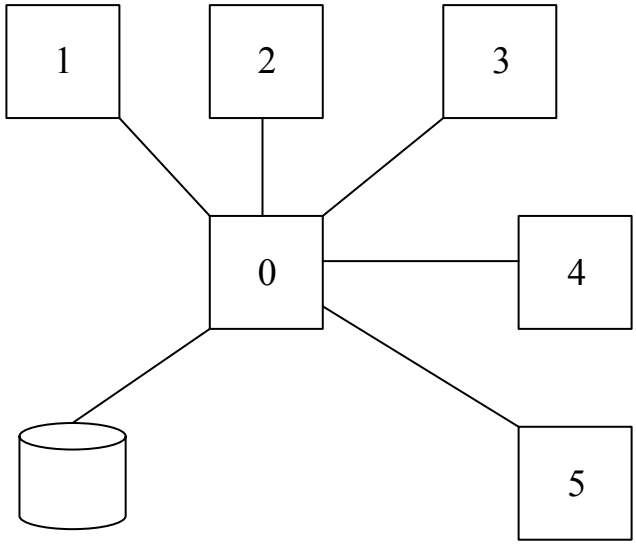
Перевірив:
Корочкін О. В.

Київ – 2015 р.

ТЕМА: Програмування для комп'ютерних систем з локальною пам'яттю. Бібліотека MPI.

Розробити програму для розв'язання в ПКС із ЛП математичної задачі:
MA = MB·MC + α·MK

Бібліотека програмування: MPI



α, MA, MB, MC, MR

Структурна схема ПКС

Виконання роботи:

Етап 1. Побудова паралельного алгоритму

1) $MA_H = MB \cdot MC_H + \alpha \cdot MK_H$

Етап 2. Розроблення алгоритмів роботи кожного процесу

Задача T0

1. Введення α, MB, MC, MK
2. **Передати** α, MB, MC_H, MK_H задачам T1-T5
3. Обчислити $MA_H = MB \cdot MC_H + \alpha \cdot MK_H$
4. **Прийняти** MA_H від задач T1-T5
5. Вивести MA

Задачі T1-T5

1. **Прийняти** α, MB, MC_H, MK_H від задачі T0
2. Обчислити $MA_H = MB \cdot MC_H + \alpha \cdot MK_H$
3. **Передати** MA_H задачі T0

Етап 4. Розроблення програми

```
1.
2.  import mpi.MPI;
3.
4.
5.  public class Executor {
6.        public static int N;
7.        public static int P;
8.        public static int H;
9.
10.       public static void main(String[] args) {
11.             P = Integer.parseInt(args[1]);
12.             N = Integer.parseInt(args[3]);
13.             H = N / P;
14.
15.             MPI.Init(args);
16.             System.out.println("Task "+ MPI.COMM_WORLD.Rank()+" started");
17.
18.             int[][] sendBuf = new int[6 * N + 2 * (H * 6) + 6][N];
19.
20.             int[][] recvBuf = new int[N + H + H + 1][N];
21.
22.             int[][] MC_send = new int[N][N];
23.             int[][] MK_send = new int[N][N];
24.             int[][] MB_send = new int[N][N];
25.             int alfa;
26.
27.             int[][] MC_recv = new int[H][N];
28.             int[][] MK_recv = new int[H][N];
29.             int[][] MB_recv = new int[N][N];
30.
31.             int[][] MA_send = new int[H][N];
32.             int[][] MA_resv = new int[N][N];
33.
34.             if (MPI.COMM_WORLD.Rank() == 0) {
35.                   //1.   Введення α, MB, MC, MK
36.                   for (int i = 0; i < N; i++) {
37.                         for (int j = 0; j < N; j++) {
38.                               MB_send[i][j] = 1;
39.                               MC_send[i][j] = 1;
40.                               MK_send[i][j] = 1;
41.                         }
42.                   }
43.                   int i = 0;
44.                   int g = 0;
45.                   int y = 0;
46.                   for (int j = 0; j < MPI.COMM_WORLD.Size(); j++) {
47.
48.                         // write MB
49.                         for (int j2 = 0; j2 < N; j2++, i++) {
50.                               for (int k = 0; k < N; k++) {
51.                                     sendBuf[i][k] = MB_send[j2][k];
52.
53.                               }
54.
55.                         }
56.
57.                         // write mc
58.                         for (; g < H * (j + 1); g++, i++) {
59.
60.                               for (int j2 = 0; j2 < MC_send.length; j2++) {
61.                                     sendBuf[i][j2] = MC_send[g][j2];
```

```
62.
63.                                 }
64.
65.                           }
66.
67.                           // write mm
68.                           for (; y < H * (j + 1); y++, i++) {
69.                                 for (int j2 = 0; j2 < MK_send.length; j2++) {
70.                                       sendBuf[i][j2] = MK_send[y][j2];
71.
72.                                 }
73.
74.                           }
75.                           //write alpha
76.                           sendBuf[i][0] = 1;
77.                           i++;
78.
79.                     }
80.
81.
82.               }
83.
84.               //2.  Передати/Прийняти α, MB, MCH, MKH
85.               MPI.COMM_WORLD.Scatter(sendBuf, 0, N + 2 * H + 1, MPI.OBJECT,
   recvBuf, 0, N + 2
86.                           * H + 1, MPI.OBJECT, 0);
87.
88.               int i = 0;
89.
90.               // get MB
91.               for (; i < MB_recv.length; i++) {
92.                     for (int j = 0; j < MB_recv[i].length; j++) {
93.                           MB_recv[i][j] = recvBuf[i][j];
94.                     }
95.               }
96.
97.               // getMC
98.               for (int j = 0; j < MC_recv.length; j++, i++) {
99.                     for (int k = 0; k < MC_recv[j].length; k++) {
100.                          MC_recv[j][k] = recvBuf[i][k];
101.                    }
102.              }
103.
104.              // getMK
105.              for (int j = 0; j < MK_recv.length; j++, i++) {
106.                    for (int k = 0; k < MK_recv[j].length; k++) {
107.                          MK_recv[j][k] = recvBuf[i][k];
108.                    }
109.              }
110.              //get alpha
111.              alfa = recvBuf[i][0];
112.
113.              //3.  Обчислити MAH = MB·MCH + α·MKH
114.              for (int j = 0; j < H; j++) {
115.                    for (int k = 0; k < N; k++) {
116.                          MA_send[j][k] = 0;
117.                          for (int m = 0; m < N; m++) {
118.                                MA_send[j][k] += MC_recv[j][m] *
   MB_recv[m][k];
119.                          }
120.                          MA_send[j][k] += MK_recv[j][k] * alfa;
121.                    }
122.              }
123.              //4.  Прийняти/Передати MAH
```

```
124.                    MPI.COMM_WORLD.Gather(MA_send, 0, H, MPI.OBJECT, MA_resv, 0,
    H, MPI.OBJECT, 0);
125.
126.                    if (MPI.COMM_WORLD.Rank() == 0) {
127.                          System.out.println("Result");
128.                          //5.   Вивести MA
129.                          matrixOutput(MA_resv);
130.                    }
131.
132.                    System.out.println("Task "+ MPI.COMM_WORLD.Rank()+"
    finished");
133.                    MPI.Finalize();
134.
135.              }
136.
137.              public static void matrixOutput(int[][] array) {
138.                    for (int i = 0; i < array.length; i++) {
139.                          for (int j = 0; j < array[i].length; j++) {
140.                                System.out.print(array[i][j] + ", ");
141.                          }
142.                          System.out.println();
143.                    }
144.                    System.out.println();
145.              }
146.
147.       }
```