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



Структурна схема ПКС

**Виконання роботи:**

**Етап 1. Побудова паралельного алгоритму**

1. MAH = MB∙MCH + α∙MKH

**Етап 2. Розроблення алгоритмів роботи кожного процесу**

**Задача Т0**

1. Введення α, MB, MC, MK
2. **Передати** α, MB, MCH, MKH задачам Т1-Т5
3. Обчислити MAH = MB∙MCH + α∙MKH
4. **Прийняти** MAH від задач Т1-Т5
5. Вивести МА

**Задачі Т1-Т5**

1. **Прийняти** α, MB, MCH, MKH від задачі Т0
2. Обчислити MAH = MB∙MCH + α∙MKH
3. **Передати** MAH задачі Т0

Етап 4. Розроблення програми

1. **import** mpi.MPI;
2. **public** **class** Executor {
3. **public** **static** **int** *N*;
4. **public** **static** **int** *P*;
5. **public** **static** **int** *H*;
6. **public** **static** **void** **main**(String[] args) {
7. *P* = Integer.*parseInt*(args[1]);
8. *N* = Integer.*parseInt*(args[3]);
9. *H* = *N* / *P*;
10. MPI.*Init*(args);
11. System.***out***.println("Task "+ MPI.*COMM\_WORLD*.Rank()+" started");
13. **int**[][] **sendBuf** = **new** **int**[6 \* *N* + 2 \* (*H* \* 6) + 6][*N*];
14. **int**[][] **recvBuf** = **new** **int**[*N* + *H* + *H* + 1][*N*];
15. **int**[][] **MC\_send** = **new** **int**[*N*][*N*];
16. **int**[][] **MK\_send** = **new** **int**[*N*][*N*];
17. **int**[][] **MB\_send** = **new** **int**[*N*][*N*];
18. **int** **alfa**;
19. **int**[][] **MC\_recv** = **new** **int**[*H*][*N*];
20. **int**[][] **MK\_recv** = **new** **int**[*H*][*N*];
21. **int**[][] **MB\_recv** = **new** **int**[*N*][*N*];
22. **int**[][] **MA\_send** = **new** **int**[*H*][*N*];
23. **int**[][] **MA\_resv** = **new** **int**[*N*][*N*];
24. **if** (MPI.*COMM\_WORLD*.Rank() == 0) {
25. //1. Введення α, MB, MC, MK
26. **for** (**int** **i** = 0; i < *N*; i++) {
27. **for** (**int** **j** = 0; j < *N*; j++) {
28. MB\_send[i][j] = 1;
29. MC\_send[i][j] = 1;
30. MK\_send[i][j] = 1;
31. }
32. }
33. **int** **i** = 0;
34. **int** **g** = 0;
35. **int** **y** = 0;
36. **for** (**int** **j** = 0; j < MPI.*COMM\_WORLD*.Size(); j++) {
37. // write MB
38. **for** (**int** **j2** = 0; j2 < *N*; j2++, i++) {
39. **for** (**int** **k** = 0; k < *N*; k++) {
40. sendBuf[i][k] = MB\_send[j2][k];
41. }
42. }
43. // write mc
44. **for** (; g < *H* \* (j + 1); g++, i++) {
45. **for** (**int** **j2** = 0; j2 < MC\_send.length; j2++) {
46. sendBuf[i][j2] = MC\_send[g][j2];
47. }
48. }
49. // write mm
50. **for** (; y < *H* \* (j + 1); y++, i++) {
51. **for** (**int** **j2** = 0; j2 < MK\_send.length; j2++) {
52. sendBuf[i][j2] = MK\_send[y][j2];
53. }
54. }
55. //write alpha
56. sendBuf[i][0] = 1;
57. i++;
58. }
59. }
60. //2. Передати/Прийняти α, MB, MCH, MKH
61. MPI.*COMM\_WORLD*.Scatter(sendBuf, 0, *N* + 2 \* *H* + 1, MPI.*OBJECT*, recvBuf, 0, *N* + 2
62. \* *H* + 1, MPI.*OBJECT*, 0);
63. **int** **i** = 0;
64. // get MB
65. **for** (; i < MB\_recv.length; i++) {
66. **for** (**int** **j** = 0; j < MB\_recv[i].length; j++) {
67. MB\_recv[i][j] = recvBuf[i][j];
68. }
69. }
70. // getMC
71. **for** (**int** **j** = 0; j < MC\_recv.length; j++, i++) {
72. **for** (**int** **k** = 0; k < MC\_recv[j].length; k++) {
73. MC\_recv[j][k] = recvBuf[i][k];
74. }
75. }
76. // getMK
77. **for** (**int** **j** = 0; j < MK\_recv.length; j++, i++) {
78. **for** (**int** **k** = 0; k < MK\_recv[j].length; k++) {
79. MK\_recv[j][k] = recvBuf[i][k];
80. }
81. }
82. //get alpha
83. alfa = recvBuf[i][0];
84. //3. Обчислити MAH = MB∙MCH + α∙MKH
85. **for** (**int** **j** = 0; j < *H*; j++) {
86. **for** (**int** **k** = 0; k < *N*; k++) {
87. MA\_send[j][k] = 0;
88. **for** (**int** **m** = 0; m < *N*; m++) {
89. MA\_send[j][k] += MC\_recv[j][m] \* MB\_recv[m][k];
90. }
91. MA\_send[j][k] += MK\_recv[j][k] \* alfa;
92. }
93. }
94. //4. Прийняти/Передати MAH
95. MPI.*COMM\_WORLD*.Gather(MA\_send, 0, *H*, MPI.*OBJECT*, MA\_resv, 0, *H*, MPI.*OBJECT*, 0);
96. **if** (MPI.*COMM\_WORLD*.Rank() == 0) {
97. System.***out***.println("Result");
98. //5. Вивести МА
99. *matrixOutput*(MA\_resv);
100. }
101. System.***out***.println("Task "+ MPI.*COMM\_WORLD*.Rank()+" finished");
102. MPI.*Finalize*();
103. }
105. **public** **static** **void** **matrixOutput**(**int**[][] array) {
106. **for** (**int** **i** = 0; i < array.length; i++) {
107. **for** (**int** **j** = 0; j < array[i].length; j++) {
108. System.***out***.print(array[i][j] + ", ");
109. }
110. System.***out***.println();
111. }
112. System.***out***.println();
113. }
114. }