Лабораторная работа №8

по курсу:

«Паралельные и распределённые вычисления»

Тема: «MPI»

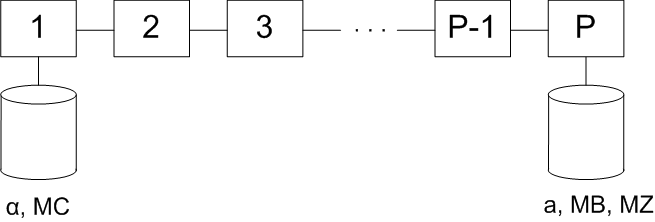
Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Воробйов Виталий

**Техническое задание**

а = max(MB - MC \* MZ \* α)



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. ai = min(MBH - MC \* MZH \* α), i = , где Р – количество процессоров
2. a = min(a, ai)

**Этап 2. Разработка алгоритмов процессов (задач)**

**№ Алгоритм задач rank < P/2-1**

1. **Если** rank = 0 – Ввод α, МС; **иначе** принять α, МС от задачи слева
2. Передать α, МС задаче справа
3. Принять MB rank \*H, MZ rank \*H от задачи справа
4. **Если** rank > 0 - передать MB (rank-1) \*H, MZ (rank -1) \*H задаче слева
5. Счёт arank = min(MBH - MC \* MZH \* α)
6. **Если** rank > 0 – принять arank+1
7. **Если** rank > 0 – Счёт arank = min(arank, arank+1)
8. **Если** rank < P - 1 – передать arank

**№ Алгоритм задач rank > P/2**

1. **Если** rank = P - 1 – Ввод МВ, МZ; **иначе** принять МВrank \*H, МZrank \*H от задачи справа
2. Передать MB (rank-1) \*H, MZ (rank -1) \*H задаче слева
3. Принять α, МС от задачи слева
4. **Если** rank < P - 1 - передать α, МС задаче справа
5. Счёт arank = min(MBH - MC \* MZH \* α)
6. **Если** rank > 0 – принять arank+1
7. **Если** rank > 0 – Счёт arank = min(arank, arank+1)
8. **Если** rank < P - 1 – передать arank
9. **Если** rank = P - 1 – Вывод а.

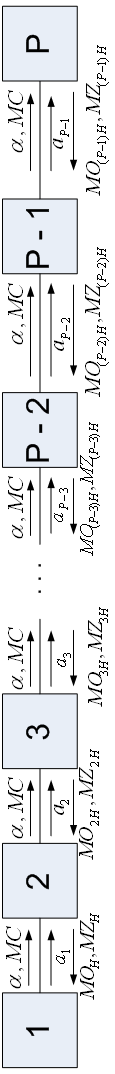
**№ Алгоритм задачи № P/2 - 1**

1. Принять α, МС от задачи слева
2. Передать α, МС задаче справа
3. Принять МВrank \*H, МZrank \*H от задачи справа
4. Передать MB (rank-1) \*H, MZ (rank -1) \*H задаче слева

**№ Алгоритм задачи № P/2**

1. Принять МВrank \*H, МZrank \*H от задачи справа
2. Принять α, МС от задачи слева
3. Передать α, МС задаче справа
4. Передать MB (rank-1) \*H, MZ (rank -1) \*H задаче слева

**Этап 3. Разработка схемы взаимодействия задач**



**Этап 4. Разработка программы.**

**Листинг программы**

**import** mpi.\*;

**public** **class** MPI\_main {

**public** **static** **void** main(String[] args) **throws** Exception {

MPI.*Init*(args);

**int** rank = MPI.*COMM\_WORLD*.Rank();

**int** size = MPI.*COMM\_WORLD*.Size();

**int** P = size;

**int** N = 4; //вводится с консоли при запуске приложения.

**int** H = N/P;

**int** currentH\_MB\_MZ = (rank + 1) \* H;

**int**[] MB = **new** **int**[currentH\_MB\_MZ \* N];

**int**[] MZ = **new** **int**[currentH\_MB\_MZ \* N];

**int**[] MC = **new** **int**[N \* N];

**int**[] alfax = { 0 };

**int**[] a = { Integer.*MIN\_VALUE* };

//////////////////////////////////////////////////////////

////////////////////////ВВОД ДАННЫХ///////////////////////

//////////////////////////////////////////////////////////

**if** (rank < P/2 - 1) {

**if** (rank == 0) {

//если НОМЕР ЗАДАЧИ 0 -> ввод alfa, МC

//ввод данных

**for** (**int** i = 0; i < N; i++) {

**for** (**int** j = 0; j < N; j++) {

MC[i\*N + j] = 1;

}

}

alfax[0] = 1;

} **else** {

MPI.*COMM\_WORLD*.Recv(alfax, 0, 1, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Recv(MC, 0, N \* N, MPI.*INT*, rank - 1, rank - 1);

}

MPI.*COMM\_WORLD*.Send(alfax, 0, 1, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Send(MC, 0, N \* N, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(MB, 0, MB.length, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(MZ, 0, MZ.length, MPI.*INT*, rank + 1, rank);

**if** (rank > 0) {

MPI.*COMM\_WORLD*.Send(MB, H \* N, MB.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Send(MZ, H \* N, MZ.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

}

} **else** **if** (rank > P / 2) {

**if** (rank == P - 1) {

//ввод данных

**for** (**int** i = 0; i < N; i++) {

**for** (**int** j = 0; j < N; j++) {

MB[i\*N + j] = 1;

MZ[i\*N + j] = 1;

}

}

} **else** {

MPI.*COMM\_WORLD*.Recv(MB, 0, MB.length, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(MZ, 0, MZ.length, MPI.*INT*, rank + 1, rank);

}

MPI.*COMM\_WORLD*.Send(MB, H \* N, MB.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Send(MZ, H \* N, MZ.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Recv(alfax, 0, 1, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Recv(MC, 0, N \* N, MPI.*INT*, rank - 1, rank - 1);

**if** (rank < P-1) {

MPI.*COMM\_WORLD*.Send(alfax, 0, 1, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Send(MC, 0, N \* N, MPI.*INT*, rank + 1, rank);

}

}

**if** (rank == P/2 - 1) {

MPI.*COMM\_WORLD*.Recv(alfax, 0, 1, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Recv(MC, 0, N \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Send(alfax, 0, 1, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Send(MC, 0, N \* N, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(MB, 0, MB.length, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(MZ, 0, MZ.length, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Send(MB, H \* N, MB.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Send(MZ, H \* N, MZ.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

}

**if** (rank == P/2) {

MPI.*COMM\_WORLD*.Recv(MB, 0, MB.length, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(MZ, 0, MZ.length, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Recv(alfax, 0, 1, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Recv(MC, 0, N \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Send(alfax, 0, 1, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Send(MC, 0, N \* N, MPI.*INT*, rank + 1, rank);

MPI.*COMM\_WORLD*.Send(MB, H \* N, MB.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

MPI.*COMM\_WORLD*.Send(MZ, H \* N, MZ.length - H \* N, MPI.*INT*, rank - 1, rank - 1);

}

//////////////////////////////////////////////////////////

/////////////////РАССПРОСТРАНЕНИЕ ДАННЫХ//////////////////

//////////////////////////////////////////////////////////

//a = max(MB - MC \* MZ \* alfa)

**int** sum;

**for** (**int** z = 0; z < H; z++) {

**for** (**int** i = 0; i < N; i++) {

sum = 0;

**for** (**int** j = 0; j < N; j++) {

sum += MC[i\*N + j] \* MZ[z\*N + j] \* alfax[0];

}

sum = MB[z\*N + i] - sum;

**if** (a[0] < sum) {

a[0] = sum;

}

}

}

//////////////////////////////////////////////////////////

/////////////////ОТПРАВЛЕНИЕ РЕЗУЛЬТАТА///////////////////

//////////////////////////////////////////////////////////

**if** (rank != P - 1) {

**int**[] b = { 0 };

MPI.*COMM\_WORLD*.Recv(b, 0, 1, MPI.*INT*, rank + 1, rank + 1);

**if** (a[0] < b[0]) {

a[0] = b[0];

}

}

**if** (rank != 0) {

MPI.*COMM\_WORLD*.Send(a, 0, 1, MPI.*INT*, rank - 1, rank);

}

**if** (rank == P - 1) {

System.*out*.println("RESULTff IS " + a[0]);

}

System.*out*.println("end " + rank);

MPI.*Finalize*();

}

}