22 апреля 2022 г. 15:22

Схема работы алгоритма следующая, изначально каждому из процессов для обработки достаются столбцы [m_begin; m_end) где m_begin = M/proc_num*proc_rank, m_end=M/proc_num*(proc_rank+1) . Далее каждыq процесс работает в соответствии с примером из таблицы.

Для proc_rank=1:

1) Начинаем прием сообщения от процесса с proc_rank = 0 с помощью неблокирующей функции Irecv и считаем значение для U[0][2] из н.у.

2) Считаем значение для U[0][3] из н.у. и посчитанное значение отправляем с помощью неблокирующей функции Isend на процессор с proc_rank=2 и, если прием Irecv из п.1 не закончил выполнение - ожидаем до конца приема сообщения

3) Снова начинаем прием с помощью неблокирующей функции Irecv сообщения от процесса с proc_rank = 0. На данном этапе мы приняли значение U[0][1] от процесса с proc_rank=0, поэтому мы считаем U[1][2] = F(U[0][1], U[0][2]), где U[0][1] была вычислена на этапе 2 на процессе с proc_rank = 0 и принята в виде сообщения на нашем процессе, а U[0][2] была вычислена на этапе 1 данного процесса.

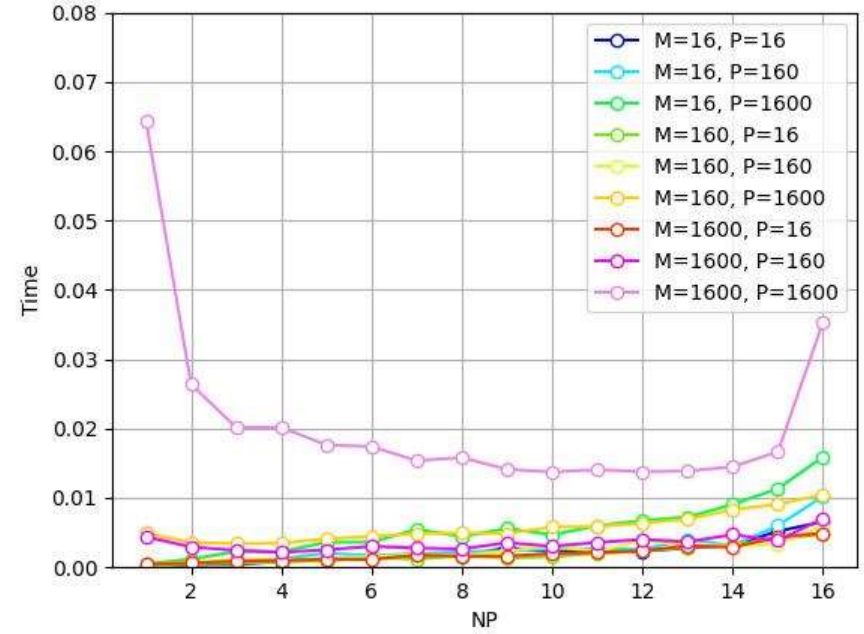
И т.д.

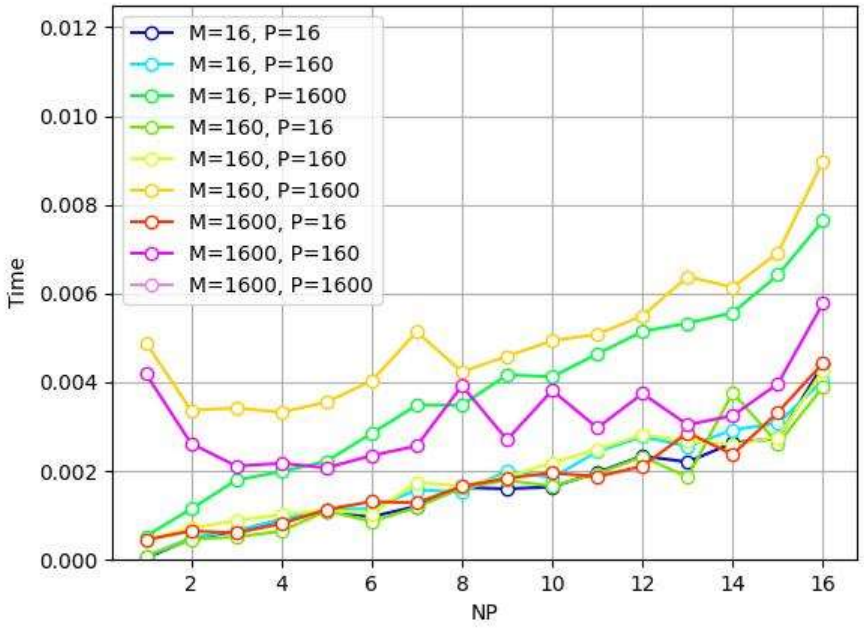
Аналогично таблица читается для процессов с proc_rank = 0 и 2.

2	5	6 (3, 4) Isend	Irecv 5 (4, 3)	6 (3, 4) Isend&wait(Irecv 5)	Irecv 5 (4, 3)	6 (3, 4) wait(Irecv 5)
1	3	4 (1, 2) Isend	Irecv 3 (2, 1)	4 (1, 2) Isend&wait(Irecv 3)	Irecv 3 (2, 1)	4 (1, 2) wait(Irecv 3)
0	1	2 Isend	Irecv 1	2 Isend&wait(Irecv 1)	Irecv 1	2 wait(Irecv 1)
p/m	0	1	2	3	4	5

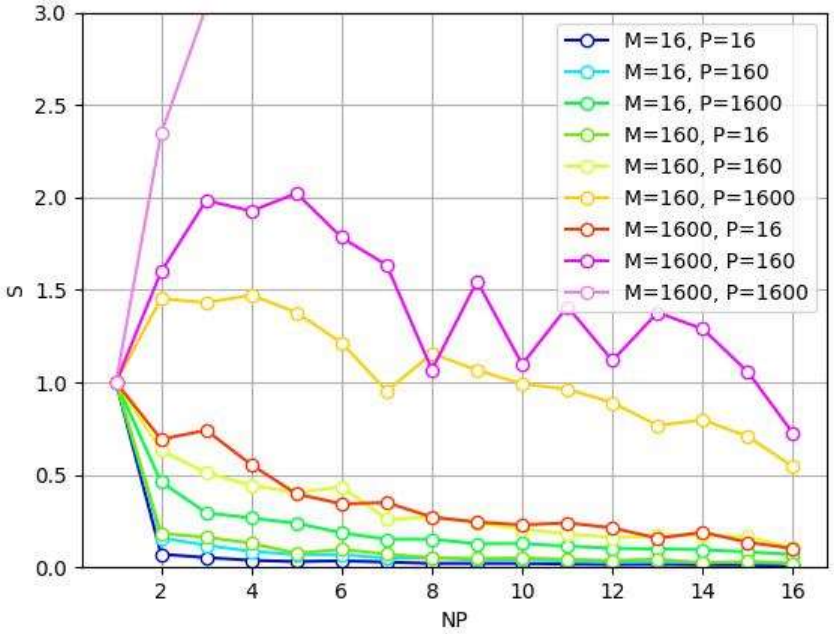
M = 6, P = 3, NP = 3: proc_rank = 0 proc_rank = 1 proc_rank = 2

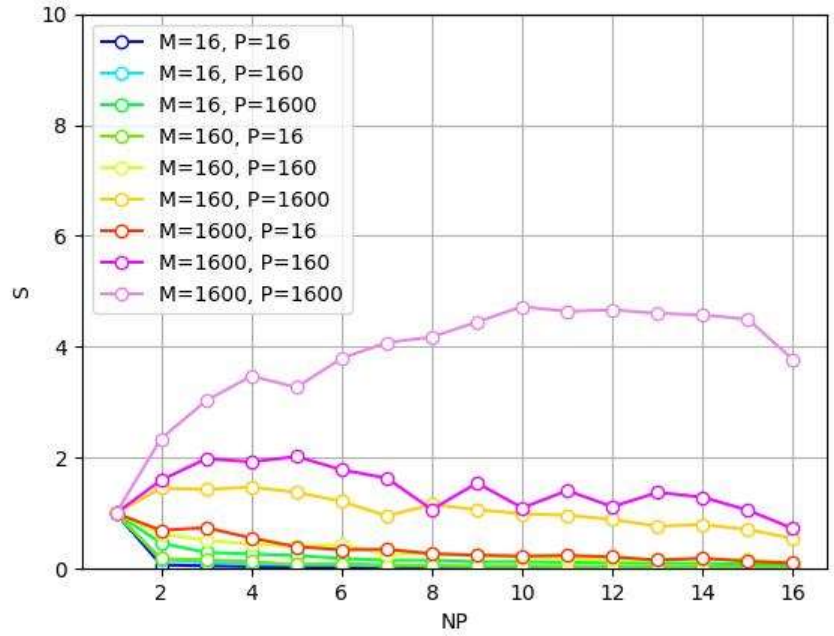
Графики зависимости времени работы программы от количества процессов



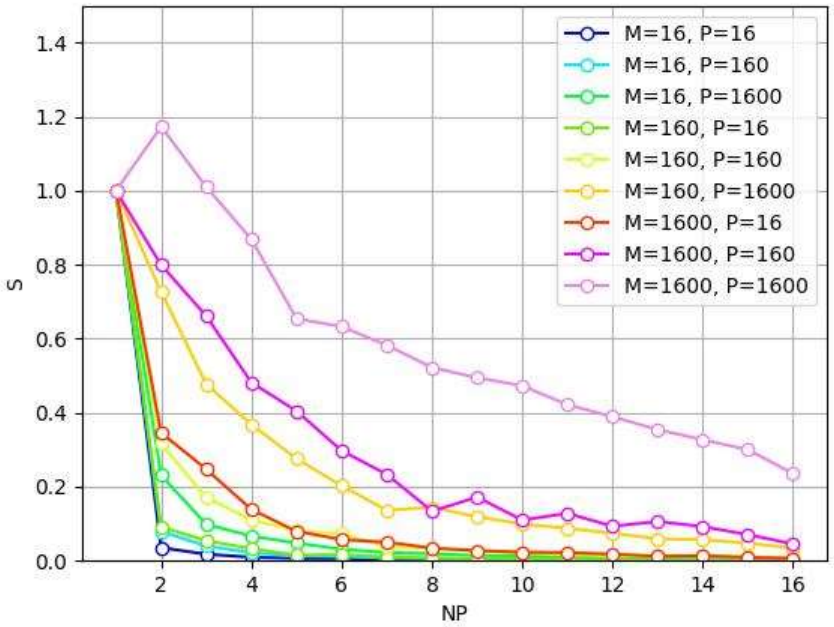


Графики зависимости ускорения от NP





Графики зависимости эффективности алгоритма от количества процессов (не нравится пик в 1.2)


M=16, P=16
M=16, P=160
M=16, P=1600
M=160, P=16
M=160, P=160
M=160, P=1600
M=1600, P=16
M=1600, P=160
M=1600, P=1600
S
NP
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
2
4
6
8
10
12
14
16