

ФИЛИАЛ МОСКОВСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
ИМ. М.В. ЛОМОНОСОВА В Г. СЕВАСТОПОЛЬ
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ

Направление подготовки: «Прикладная математика и информатика»

ОТЧЁТ
по практической работе №1
дисциплины «Распределенные системы»

Работу выполнил:
Студент 4 курса
Группы ПМ-401
Воронец Владимир

Севастополь, 2024

ПОСТАНОВКА ЗАДАЧИ

В транспьютерной матрице размером 4×4 , в каждом узле которой находится один процесс, необходимо выполнить операцию сбора данных (длиной 4 байта) от всех процессов (`MPI_ALLGATHER`). Данные, посылаемые i -ым процессом, помещаются в i -ый элемент результирующего буфера каждого процесса. После завершения операции содержимое результирующих буферов у всех процессов должно быть одинаково.

Реализовать программу, моделирующую выполнение операции `MPI_ALLGATHER` на транспьютерной матрице при помощи пересылок `MPI` типа точка-точка.

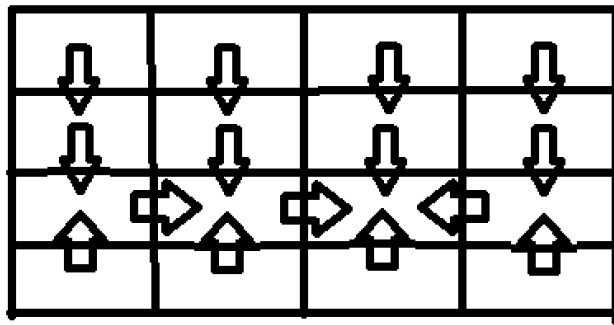
Получить временную оценку работы алгоритма. Оценить сколько времени потребуется для выполнения операции `MPI_ALLGATHER`, если все процессы выдали ее одновременно. Время старта равно 100, время передачи байта равно 1 ($T_s=100, T_b=1$). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

РЕАЛИЗАЦИЯ АЛГОРИТМА

В каждой ячейке с номером i изначально лежит только i -й фрагмент буфера.

Шаг 1

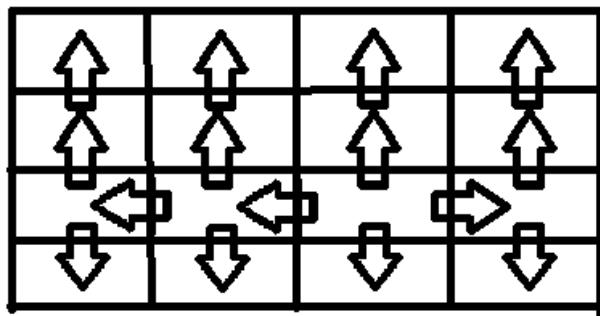
- С помощью команды `MPI_Recv` собираем в буфер все данные от входных процессов;
- Добавляем к ним значение i -го элемента буфера;
- Отправляем дальше с помощью команды неблокирующей отправки сообщения `MPI_Isend`;
- Данные передаются в центральный процесс матрицы (с координатами $[2, 2]$) в соответствии с топологией на рисунке, приведённом ниже;



- В результате первого шага в центральном узле матрицы будет полностью собранный результирующий буфер.

Шаг 2

- На этом этапе необходимо разослать имеющийся в узле с координатами [2, 2] результирующий буфер всем процессам. Для этого были разосланы значения буферов в обратном первому шагу порядке.



В результате работы алгоритма содержимое результирующих буферов у всех процессов становится одинаковым:

```

14 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
10 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
11 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 7 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 6 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
15 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 2 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 9 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 8 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
13 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 5 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 4 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
12 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 1 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 0 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 3 process recv buf: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

КОМАНДЫ ДЛЯ ЗАПУСКА

```
mpicc MPI_ALLGATHER.c -o ./MPI_ALLGATHER
```

```
mpiexec --oversubscribe -np 25 ./MPI_ALLGATHER
```

ВРЕМЕННАЯ ОЦЕНКА

Время работы алгоритма оцениваем по самому “далёкому” в топологии процессу. Для него требуется 4 пересылки на первом шаге и столько же на втором. Так как размер буфера равен

$16 * \text{sizeof}$ байта, $T_s = 100$, $T_b = 1$, получим итоговую оценку

$$8 * (T_s + 64 * T_b) = 1312 \text{ мс}$$