

[13] ✓ 0 cek. !mpirun --allow-run-as-root -np 1 ./task1 1000000

✓ TASK 1 – Stats (Scatterv + Reduce)  
N = 1000000, processes = 1  
Mean = 0.500456  
Stddev = 0.288574  
Execution time: 0.0300999 seconds

[15] ▶ !mpirun --allow-run-as-root -np 1 ./task2 512

✓ ... TASK 2 – Gaussian elimination (Scatter + Bcast + Gather)  
N = 512, processes = 1  
Execution time: 0.039413 seconds  
x[0..9]:  
x[0] = 0.00149792  
x[1] = 0.00331129  
x[2] = -0.000342662  
x[3] = -0.00135669  
x[4] = -0.00111961  
x[5] = 0.000531986  
x[6] = 0.000175495  
x[7] = -0.00114454  
x[8] = 0.00195408  
x[9] = 0.00150617

[16]

[16]  
✓ 1 сек.

```
!mpic++ -O3 -std=c++17 task3_floyd_allgather.cpp -o task3
```

[17]  
✓ 0 сек.

```
!mpirun --allow-run-as-root -np 1 ./task3 256
```



TASK 3 – Floyd-Warshall (Scatter + Allgather)

N = 256, processes = 1

Execution time: 0.0336727 seconds

Top-left 10x10 of distance matrix:

```
0 5 4 3 4 5 6 2 7 7
4 0 4 4 5 5 3 4 4 4
3 5 0 5 3 1 3 5 5 4
5 4 5 0 5 4 4 4 4 6
3 5 4 5 0 4 2 5 5 5
5 6 3 7 2 0 4 6 7 4
1 4 5 4 4 6 0 3 6 5
4 3 5 1 4 5 4 0 5 5
5 4 4 4 4 4 4 3 0 4
6 6 6 7 6 5 6 6 8 0
```

Ответы на контрольные вопросы

1) Как изменяется время выполнения при увеличении числа процессов? Почему?

Обычно время уменьшается, потому что работа делится на большее число процессов (параллелизм). Но после некоторого числа процессов ускорение замедляется или исчезает из-за:

- роста коммуникаций (Scatter/Reduce/Bcast/Allgather),
- синхронизаций (ожидание других процессов),
- уменьшения вычислений на процесс (слишком мелкие куски данных).

2) Какие факторы влияют на производительность?

- объём коммуникаций и их частота,
- задержки сети и пропускная способность,
- баланс нагрузки (равномерность распределения данных),
- локальность данных/кэш/память,
- стоимость коллективных операций (Bcast, Reduce, Allgather),
- размер задачи (для маленьких N параллелизм не окупается).

3) Как оптимизировать передачу данных между процессами?

- уменьшать число коллективных операций,

- передавать только необходимую часть данных,
- использовать неблокирующие операции (MPI\_Isend/Irecv, MPI\_Ibcast, MPI\_Iallgather),
- перекрывать коммуникации вычислениями,
- выбирать правильную схему разбиения данных (чтобы меньше обмениваться).

#### 4) Какие ограничения при работе с большими данными?

- память: матрицы  $N \times N$  могут не помещаться (особенно Флойд–Уоршелл),
- коммуникации становятся доминирующими (особенно Allgather),
- масштабируемость ограничена сетью и синхронизациями,
- возможны переполнения типов, если считать суммы/индексы в int без контроля,
- I/O и вывод результатов становятся узким местом.