

Предмет: Параллельные методы суперкомпьютерных вычислений

Задание: Практическое задание номер 1 (Стереотипы параллельного программирования)

ФИО группа: Васильев Иван Николаевич Б05-104а

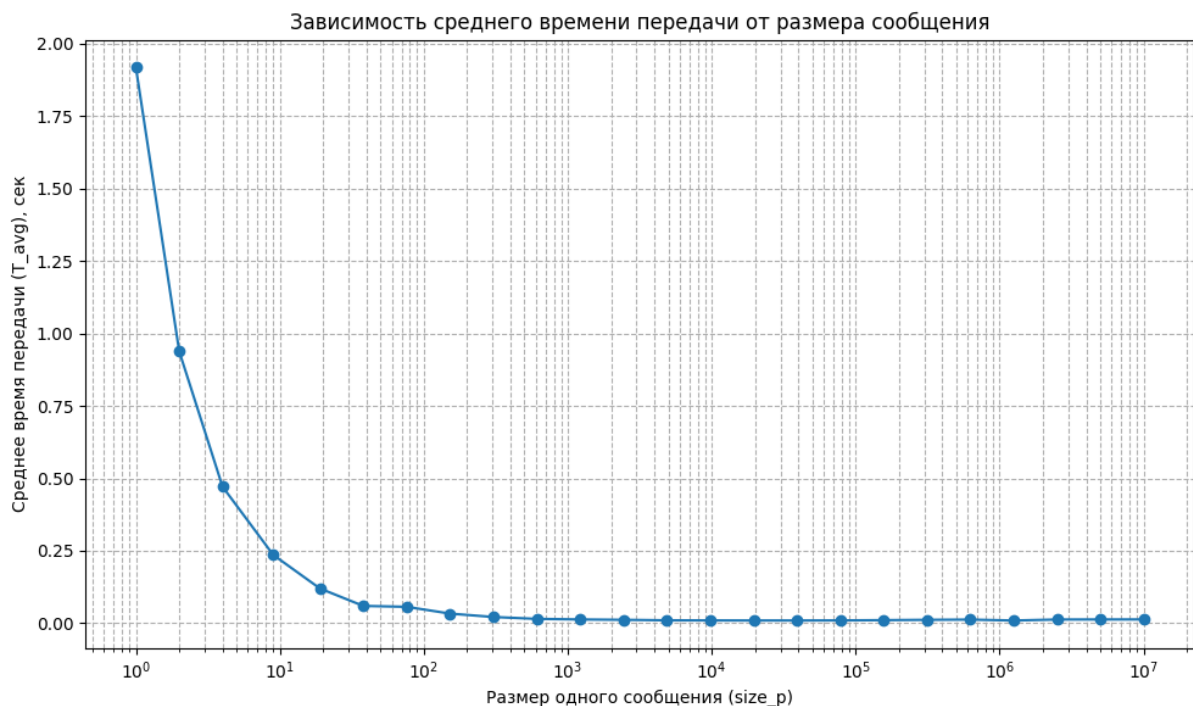
Подзадача 1: О выгоды асинхронных обменов

mode=blocking size=10000000 procs=1 time=14.091666
mode=blocking size=10000000 procs=2 time=14.646478
mode=blocking size=10000000 procs=4 time=15.122638
mode=blocking size=10000000 procs=8 time=17.076852
mode=blocking size=10000000 procs=16 time=17.832972
mode=blocking size=10000000 procs=24 time=18.830070

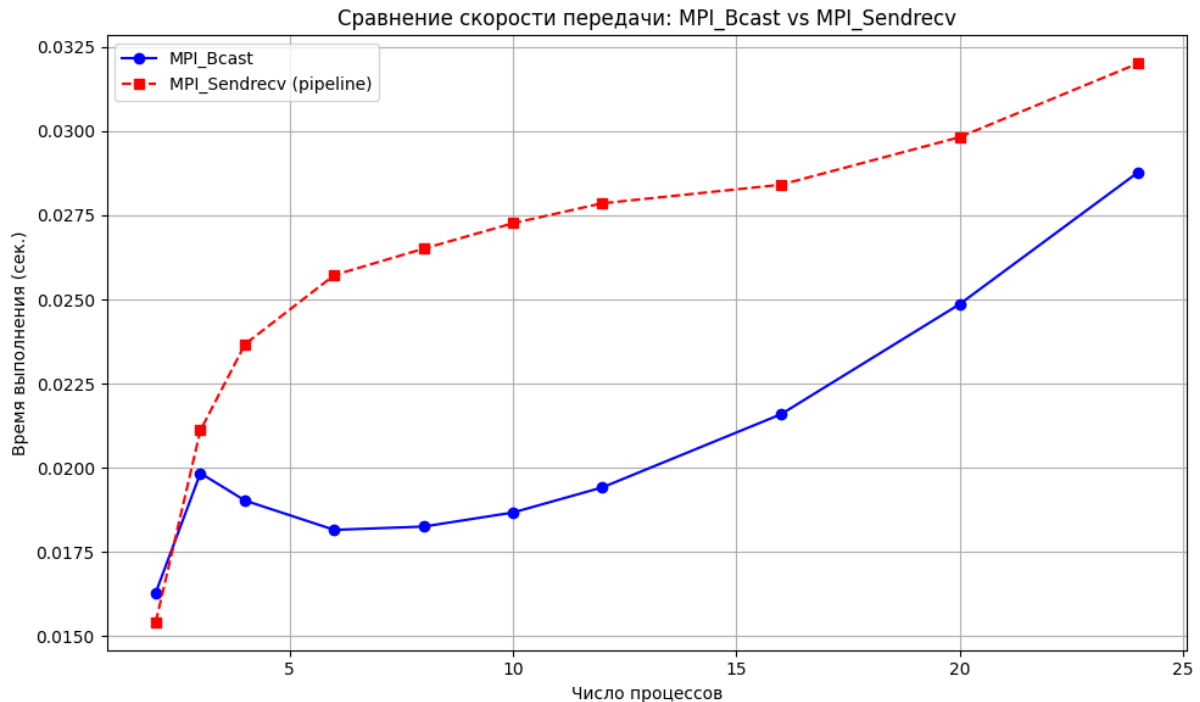
mode=nonblocking size=10000000 procs=1 time=14.010263
mode=nonblocking size=10000000 procs=2 time=14.704928
mode=nonblocking size=10000000 procs=4 time=15.530473
mode=nonblocking size=10000000 procs=8 time=17.649632
mode=nonblocking size=10000000 procs=16 time=20.380231
mode=nonblocking size=10000000 procs=24 time=18.353736

Как показывают данные эксперименты, асинхронные обмены не приносят преимуществ в производительности, возможно дело в том, что издержки на коммуникации между процессами перевешивают выгоду от параллелизма, или size=10000000 элементов — это может быть слишком мало для параллельной эффективности на большом числе потоков.

Подзадача 2: Как зависит скорость передачи сообщения от его длины?



Подзадача 3: Сравнить скорости коллективный обменов и последовательных блокирующих пересылок



1. Поведение при малом числе процессов (2)

Конвейерная схема (pipe) оказывается чуть быстрее, возможно, из-за отсутствия накладных расходов на оптимизированную реализацию MPI_Bcast.

Также возможно, что pipe из-за простой схемы передачи быстрее только при 2-х процессах.

2. Поведение при увеличении числа процессов

Время выполнения MPI_Bcast растёт медленно (почти логарифмически) — признак оптимизированной реализации (часто реализуется как дерево).

Время выполнения pipe растёт линейно — каждое сообщение передаётся последовательно от одного процесса к следующему, и каждый следующий процесс ждёт завершения предыдущего (строго последовательный характер, нет параллелизма).

MPI_Bcast масштабируется значительно лучше, чем MPI_Sendrecv, и становится эффективнее уже начиная с 3–4 процессов.

Расположение кода на кластере:

~student/25-VIN

Номер Исходник Скрипт Результат

Подзадача 1: mpi_overlap.c qs_overlap res_overlap.txt

Подзадача 2: mpi_bw.c qs_bw bw_out.txt

Подзадача 3: mpi_bcast_pipeline.c qs_bcast res_bcast.txt