

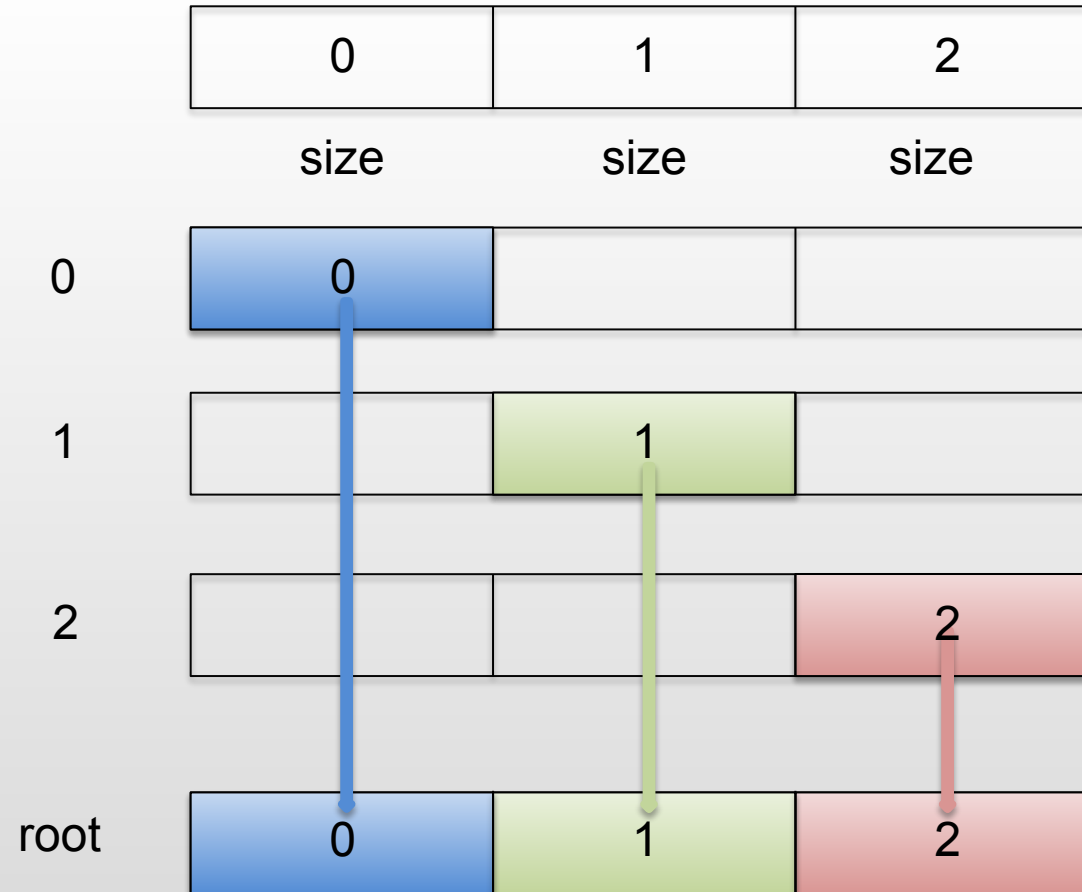
Практика

MPI-групповые операции

Часть 2 (MPI_Gather)

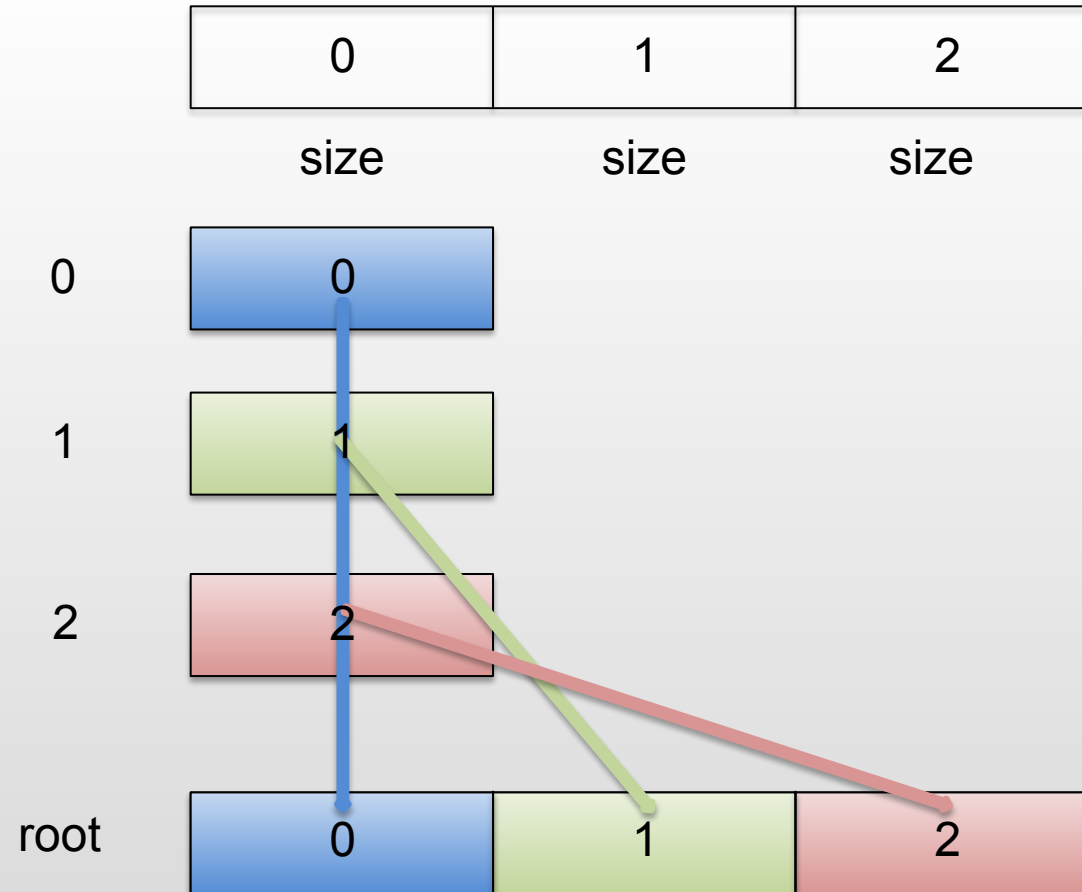
MPI – групповые операции

Сбор данных



MPI – групповые операции

Сбор данных



MPI – групповые операции

Сбор данных – MPI_Gather

Для такого сбора информации от всех процессов некоторой группы (на всех процессах одинаковое количество данных) используется функция `MPI_Gather()`.

```
#include <mpi.h>
```

```
int MPI_Gather (void *sendbuf, int sendcount,  
MPI_Datatype sendtype, void *recvbuf, int recvcount,  
MPI_Datatype recvttype, int root, MPI_Comm comm)
```

Возвращаемое значение – стандартное:

`MPI_SUCCESS` – все хорошо, иначе – что-то пошло не так.

MPI – групповые операции

Сбор данных – MPI_Gather

```
int MPI_Gather (void *sendbuf, int sendcount, MPI_Datatype sendtype,
```

sendbuf – адрес памяти, где у процесса лежат собираемые данные (у каждого процесса может быть свой)

sendcount – количество собираемых данных с процесса (должно быть одинаково для всех процессов)

sendtype – тип собираемых данных (должен быть одинаков для всех процессов)

MPI – групповые операции

Сбор данных – MPI_Gather

```
int MPI_Gather (void *sendbuf, int sendcount, MPI_Datatype sendtype,  
               void *recvbuf, int recvcount, MPI_Datatype recvtype, int root,  
               MPI_Comm comm)
```

recvbuf – адрес памяти, начиная с которого будут упорядоченно складываться собираемые данные (использует только **root**), не может совпадать с **sendbuf**.

recvcount – количество собираемых данных с процесса (использует только **root**, должно совпадать с **sendcount**)

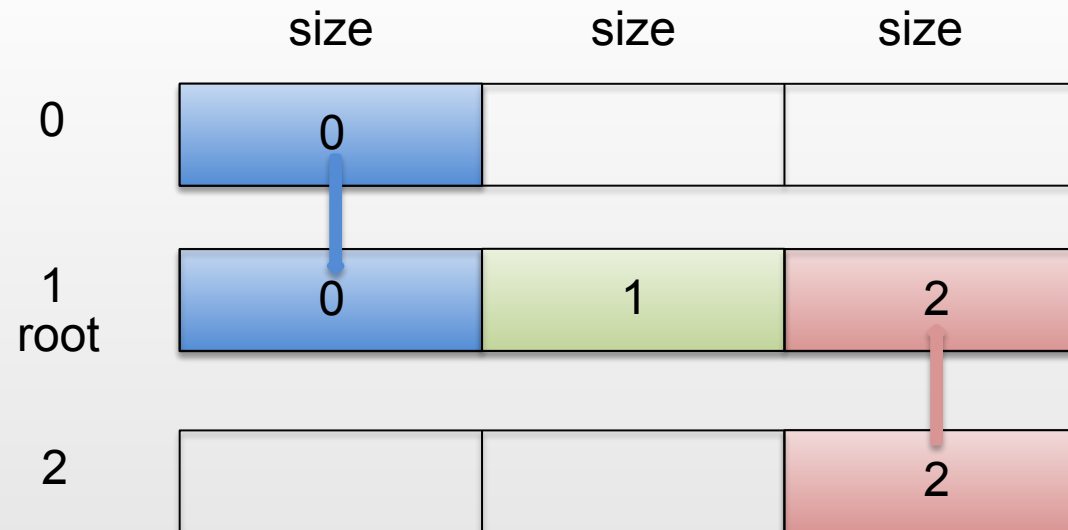
recvtype – тип собираемых данных (использует только **root**, должен совпадать с **sendtype**)

root – ранг собирающего процесса

comm – коммуникатор группы процессов

MPI – групповые операции

Сбор данных – MPI_Gather



Для такой передачи данных на процессе с рангом **root** (и только на нем!) вместо **sendbuf** указывается **MPI_IN_PLACE**

MPI – групповые операции

Сбор данных – MPI_Gather – задача

Постановка задачи

Есть массив целых чисел из $N=120$ элементов.

Заполнением массива занимаются различные процессы. Каждый процесс получает в свою зону ответственности одинаковое количество N / size элементов и заполняет их значением своего ранга. Если N на size нацело не делится – прекращаем работу MPI-приложения с сообщением об ошибке.

Далее все собирается на процессе с рангом 0 по порядку с помощью MPI_Gather и сбрасывается в отдельный файл в текстовом виде. Значения в выходном файле отделяются друг от друга пробелом.

