Министерство науки и высшего образования РФ

«Дальневосточный федеральный университет»

(ДВФУ)

Капитан В.Ю.

Лабораторная работа 3

Процедуры коллективного обмена MPI

Владивосток

2022

Коллективные обмены

При выполнении коллективного обмена сообщение пересылается от одного процесса нескольким или наоборот, один процесс собирает данные от нескольких процессов. MPI поддерживает такие виды коллективного обмена, как широковещательная передача, операции приведения (редукции), распределение и сбор данных и т. д.

Коллективные обмены характеризуются следующим:

• коллективные обмены не могут взаимодействовать с двухточечными.

Коллективная передача, например, не может быть перехвачена двухточечной подпрограммой приема;

• коллективные обмены могут выполняться как с синхронизацией, так и без нее;

• все коллективные обмены являются блокирующими для инициировавшего их

обмена;

• теги сообщений назначаются системой.

В коллективном обмене участвует каждый процесс из некоторой области взаимодействия.

Можно организовать обмен и в подмножестве процессов, для этого имеются средства создания новых областей взаимодействия и соответствующих им коммуникаторов.

Широковещательная рассылка

Широковещательная рассылка выполняется выделенным процессом, который называется главным (root). Все остальные процессы, принимающие участие в обмене, получают по одной копии сообщения от главного процесса (рис. 1).

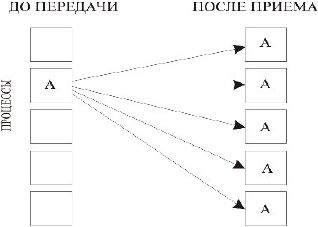


Рисунок 1. Широковещательная рассылка

Выполняется широковещательная рассылка с помощью подпрограммы MPI\_Bcast.

Операции редукции

Операции редукции относятся к категории глобальных вычислений. В глобальной операции приведения к данным от всех процессов из заданного коммуникатора применяется операция MPI\_Reduce (рис. 2).

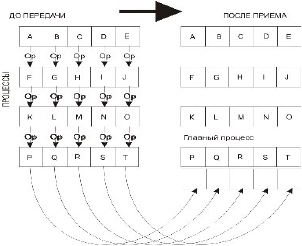


Рисунок 2. Глобальная операция приведения

Аргументом операции приведения является массив данных - по одному элементу от каждого процесса. Результат такой операции - единственное значение.

MPI\_Gather

Собирает сообщения от каждого из процессов в группе в приемный буфер процесса с рангом 'root".

Формат вызова:

MPI\_Gather ( &sendbuf, sendcount, sendtype, &recvbuf, recvcount, recvtype, root, comm )

Cледует заметить, что sendtype и recvtype, в общем случае, могут различаться, а потому будут задавать разную интерпретацию данных на приемной и передающей стороне; процесс root также отправляет данные, но в свой же приемный буфер.

MPI\_Scatter

Эта функция является обратной к функции MPI\_Gather: отдельные части передающего буфера процесса с рангом 'root& распределяются по приемным буферам всех других процессов в группе.

Формат вызова:

MPI\_Scatter ( &sendbuf, sendcount, sendtype, &recvbuf, recvcount, recvtype, root, comm )

MPI\_Allgather

Эта функция аналогична функции MPI\_Gather, за исключением того, что прием данных осуществляет не один процесс, а все процессы: каждый процесс имеет специфическое содержимое в передающем буфере, но все процессы получают в итоге одинаковое содержимое в приемном буфере.

Формат вызова:

MPI\_Allgather ( &sendbuf, sendcount, sendtype, &recvbuf, recvcount, recvtype, comm )

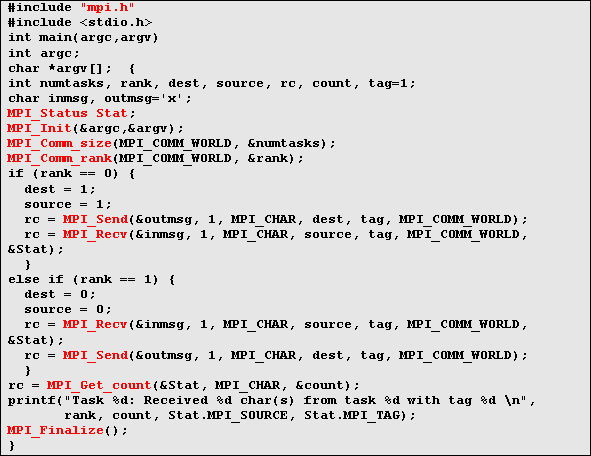
MPI\_Alltoall

Каждый процесс отдельные части своего передающего буфера рассылает всем остальным процессам; каждый процесс получает эти части от всех остальных и размещает их по порядку рангов процессов, от которых они получены.

Формат вызова:

MPI\_Alltoall ( &sendbuf, sendcount, sendtype, &recvbuf, recvcount, recvtype, comm )

В нижеследующем примере, с помощью функции MPI\_Scatter строки массива рассылаются отдельным процессам:



Создание группы процессов

Для организации коллективных обменов на подмножестве процессов создают группу и соответствующий ей коммуникатор. Группой называют упорядоченное множество процессов. Каждому процессу в группе сопоставлен свой ранг. Операции с группами могут выполняться отдельно от операций с коммуникаторами, но в операциях обмена используются только коммуникаторы. В MPI имеется специальная предопределенная пустая группа MPI\_GROUP\_EMPTY.

Коммуникаторы бывают двух типов: интракоммуникаторы - для операций внутри одной группы процессов и интеркоммуникаторы - для двухточечного обмена между двумя группами процессов.

В MPI-программах чаще используются интракоммуникаторы. Интракоммуникатор включает экземпляр группы, контекст обмена для всех его видов, а также, возможно, виртуальную топологию и другие атрибуты.

Созданию нового коммуникатора предшествует создание соответствующей группы процессов. Операции создания групп аналогичны математическим операциям над множествами:

• объединение- к процессам первой группы добавляются процессы второй группы, не принадлежащие первой;

• пересечение- в новую группу включаются все процессы, принадлежащие двум группам одновременно. Ранги им назначаются как в первой группе;

• разность- в новую группу включаются все процессы первой группы, не входящие во вторую группу. Ранги назначаются как в первой группе.

Новую группу можно создать только из уже существующих групп. Базовая группа, из которой формируются все другие группы, связана с коммуникатором MPI\_COMM\_WORLD.

Доступ к группе group, связанной с коммуникатором comm можно получить, обратившись к подпрограмме MPI\_Comm\_group.

В MPI имеются подпрограммы-конструкторы новых групп.

Есть и деструктор - MPI\_Group\_free.

Создание коммуникатора - коллективная операция и соответствующая подпрограмма должна вызываться всеми процессами коммуникатора. Подпрограмма MPI\_Comm\_dup дублирует уже существующий коммуникатор.

Подпрограмма MPI\_Comm\_create создает новый коммуникатор из подмножества процессов другого коммуникатора.

Вызов этой подпрограммы должны выполнить все процессы из старого коммуникатора, даже если они не входят в группу group, с одинаковыми аргументами. Данная операция применяется только к интракоммуникаторам. Она позволяет выделять подмножества процессов со своими областями взаимодействия, если, например, требуется уменьшить "зернистость" параллельной программы.

**Лабораторная работа**

Задание 3.1 Вычислить среднее значение массива случайных чисел с помощью MPI\_Scatter и MPI\_Gather.

Необходимо реализовать следующий алгоритм:

1) Создайте массив и заполните его случайными числами в root процессе (процесс 0).

2) Разошлите числа на все процессы, давая каждому процессу одинаковое количество чисел.

3) Каждый процесс вычисляет среднее для своего подмножества чисел.

4) Соберите все средние значения в root процесс. Root процесс затем вычисляет среднее значение этих чисел для получения окончательного среднего значения для всего массива.

Задание 3.2 Реализовать программу для рассылки частей квадратной матрицы с использованием команды MPI\_Alltoall.

Матрицу необходимо заполнить случайными значениями. Организовать вывод данных из массивов до и после применения команды MPI\_Alltoall.

Задание 3.3. Реализовать программу для рассылки данных от root процесса всем процессам с помощью MPI\_Scatter. Затем в каждом процессе просуммировать полученные данные и с помощью команды MPI\_Reduce отправить назад в root процесс и посчитать общую сумму, вывести ее на экран.