

## Лабораторная работа №3

### Обобщенная передача данных

**Цель:** изучить основные принципы обобщенной передачи данных в технологии MPI на примере использования в рамках языка C++.

Для получения **теоретических сведений** настоятельно рекомендуется при подготовке изучить материалы, представленные в списке литературы в конце разработки, а также прочие материалы по тематике лабораторной работы, представленные в открытых источниках.

Далее следует краткий конспект материала, приведенного в данных источниках, в конце включающий короткие примеры фрагментов программ.

#### 1. Обобщенная передача данных от одного процесса всем процессам

Обобщенная операция передачи данных от одного процесса всем процессам (**распределение данных**) отличается от широковещательной рассылки тем, что процесс передает процессам различающиеся данные. Выполнение данной операции может быть обеспечено при помощи функции:

```
int MPI_Scatter(void *sbuf, int scount, MPI_Datatype stype,
               void *rbuf, int rcount, MPI_Datatype rtype, int root, MPI_Comm comm);
```

где

sbuf, scount, stype - параметры передаваемого сообщения (scount определяет количество элементов, передаваемых на каждый процесс);

rbuf, rcount, rtype - параметры сообщения, принимаемого в процессах;

root – ранг процесса, выполняющего рассылку данных;

comm - коммунитор, в рамках которого выполняется передача данных.

При вызове этой функции процесс с рангом **root** произведет передачу данных всем другим процессам в коммуниторе. Каждому процессу будет отправлено **scount** элементов. Процесс с рангом 0 получит блок данных из **sbuf** из элементов с индексами от 0 до **scount-1**, процессу с рангом 1 будет отправлен блок из элементов с индексами от **scount** до **2\* scount-1** и т.д. Тем самым, общий размер отправляемого сообщения должен быть равен **scount \* p** элементов, где **p** есть количество процессов в коммуниторе **comm**.

Функция **MPI\_Scatter** определяет коллективную операцию, вызов этой функции при выполнении рассылки данных должен быть обеспечен в каждом процессе коммунитора, данная функция передает всем процессам сообщения одинакового размера.

#### 2. Обобщенная передача данных от всех процессов одному процессу

Операция обобщенной передачи данных от всех процессоров одному процессу (**сбор данных**) является обратной к процедуре распределения данных:

```
int MPI_Gather(void *sbuf, int scount, MPI_Datatype stype,
               void *rbuf, int rcount, MPI_Datatype rtype, int root, MPI_Comm comm);
```

где

sbuf, scount, stype - параметры передаваемого сообщения;

rbuf, rcount, rtype - параметры принимаемого сообщения;

root – ранг процесса, выполняющего сбор данных;

comm - коммунитор, в рамках которого выполняется передача данных.

При выполнении функции **MPI\_Gather** каждый процесс в коммуниторе передает данные из буфера **sbuf** на процесс с рангом **root**. Процесс с рангом **root** собирает все получаемые данные в буфере **rbuf** (размещение данных в буфере осуществляется в соответствии с рангами процессов-отправителей сообщений). Для того, чтобы разместить все поступающие данные, размер буфера

**rbuf** должен быть равен **scount \* p** элементов, где **p** есть количество процессов в коммуникаторе **comm**.

Функция **MPI\_Gather** также определяет коллективную операцию, и ее вызов при выполнении сбора данных должен быть обеспечен в каждом процессе коммуникатора, при использовании функции **MPI\_Gather** сборка данных осуществляется только на одном процессе. Для получения всех собираемых данных на каждом из процессов коммуникатора необходимо использовать **функцию сбора и рассылки**:

```
int MPI_Allgather(void *sbuf, int scount, MPI_Datatype stype,
void *rbuf, int rcount, MPI_Datatype rtype, MPI_Comm comm);
```

### 3. Общая передача данных от всех процессов всем процессам

Передача данных от всех процессов всем процессам является наиболее общей операцией передачи данных обеспечивается при помощи функции:

```
int MPI_Alltoall(void *sbuf,int scount,MPI_Datatype stype,
void *rbuf,int rcount,MPI_Datatype rtype,MPI_Comm comm);
```

где

sbuf, scount, stype - параметры передаваемых сообщений,  
rbuf, rcount, rtype - параметры принимаемых сообщений,  
comm - коммуникатор, в рамках которого выполняется передача данных.

При выполнении функции **MPI\_Alltoall** каждый процесс в коммуникаторе передает данные из **scount** элементов каждому процессу (общий размер отправляемых сообщений в процессах должен быть равен **scount \* p** элементов, где **p** есть количество процессов в коммуникаторе **comm**) и принимает сообщения от каждого процесса.

Вызов функции **MPI\_Alltoall** при выполнении операции общего обмена данными должен быть выполнен в каждом процессе коммуникатора.

### 4. Дополнительные операции редукции данных

Функция **MPI\_Reduce** обеспечивает получение результатов редукции данных только на одном процессе. Для получения результатов редукции данных на каждом из процессов коммуникатора необходимо использовать **функцию редукции и рассылки**:

```
int MPI_Allreduce(void *sendbuf, void *recvbuf, int count,
MPI_Datatype type, MPI_Op op, MPI_Comm comm);
```

Функция **MPI\_AllReduce** выполняет рассылку между процессами всех результатов операции редукции.

**Таблица 1. Сводный перечень коллективных операций данных**

Вид коллективной операции	Функция MPI
Передача от одного процесса всем процессам ( <b>широковещательная рассылка</b> )	<b>MPI_Bcast</b>
Сбор и обработка данных на одном процессе от всех процессов ( <b>редукция данных</b> )	<b>MPI_Reduce</b>
- то же с рассылкой результатов всем процессам	<b>MPI_Allreduce</b> <b>MPI_Reduce_scatter</b>
- то же с получением частичных результатов обработки	<b>MPI_Scan</b>
Обобщенная передача от одного процесса всем процессам ( <b>распределение данных</b> )	<b>MPI_Scatter</b> <b>MPI_Scatterv</b>

Обобщенная передача от всех процессов одному процессу ( <b>сбор данных</b> )	<b>MPI_Gather</b> <b>MPI_Gatherv</b>
- то же с рассылкой результатов всем процессам	<b>MPI_Allgather</b> <b>MPI_Allgather</b> v
Общая передача данных от всех процессов всем процессам	<b>MPI_Alltoall</b> <b>MPI_Alltoallv</b>

### Лабораторные задания

**Задание.** Модифицировать программу, написанную на Л.Р. №1 так чтобы она работала на основе обобщенной передачи сообщений. **Результаты работы сравнить (с результатами полученными в Л.Р. №2) и занести в отчет.**

### Требования к сдаче работы

1. При подготовке изучить теоретический материал по тематике лабораторной работы, представленный в списке литературы ниже, выполнить представленные примеры, занести в отчёт результаты выполнения.
2. Продемонстрировать программный код для лабораторного задания.
3. Продемонстрировать выполнение лабораторных заданий (можно в виде скриншотов).
4. Ответить на контрольные вопросы.
5. Показать преподавателю отчет.

### Литература

1. Спецификации стандарта Open MPI (версия 1.6, на английском языке):  
<http://www.open-mpi.org/doc/v1.6/>
2. Материалы, представленные на сайте intuit.ru в рамках курса «Intel Parallel Programming Professional (Introduction)»:  
<http://old.intuit.ru/department/supercomputing/ppintel/5/>
3. С.А. Лупин, М.А. Посыпкин Технологии параллельного программирования. – М.: ИД «ФОРУМ»: ИНФРА-М, 2011. – С. 12-96. (Глава, посвященная MPI)
4. Отладка приложений MPI в кластере HPC  
[http://msdn.microsoft.com/ru-ru/library/dd560808\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/dd560808(v=vs.100).aspx)
5. [http://www.parallel.ru/tech/tech\\_dev/mpi.html](http://www.parallel.ru/tech/tech_dev/mpi.html)
6. [http://msdn.microsoft.com/ru-ru/library/ee441265\(v=vs.100\).aspx#BKMK\\_debugMany](http://msdn.microsoft.com/ru-ru/library/ee441265(v=vs.100).aspx#BKMK_debugMany)