

Основы MPI

Курс «Параллельное программирование»

Евгений Юлюгин
yulyugin@gmail.com

10 марта 2014 г.

- 1 Обзор
- 2 История MPI
- 3 Базовые функции MPI
- 4 Компиляция и запуск
- 5 Конец

История MPI

- MPI расшифровывается как интерфейс передачи сообщений (*англ.* Message Passing Interface).
- Программный интерфейс, позволяющий обмениваться сообщениями между процессами выполняющими одну задачу.
- Разработан Уильямом Гроуппом, Эвином Ласком и другими.
- Первая версия разрабатывалась в 1993-1994 году.
- MPI версии 1 вышла в 1994 году.
- MPI 1.1 опубликован в 12 июня 1995 года. Поддерживается большинством современных реализаций MPI. Первая раелизация появилась в 2002 году.
- Существуют реализации для языков Fortran, Java, C и C++

Инициализация и завершение процессов

Определены в заголовочном файле `mpi.h`

```
int MPI_Init(int *pargc, char ***pargv);  
int MPI_Finalize(void);
```

MPI communicator

- `MPI_COMM_WORLD` is the initially defined universe intracommunicator for all processes to conduct various communications once `MPI_INIT` has been called,
- `MPI_COMM_SELF`,
- User defined communicators.

Self-identification

```
MPI_Comm_size(MPI_Comm comm, int *size);
```

Returns the size of the group associated with communicator.

```
MPI_Comm_rank(MPI_Comm comm, int *rank);
```

Determines the rank of the calling process in the communicator.

Send/Receive Overview

```
int MPI_Send(void *buf, int count,
             MPI_Datatype datatype, int dest, int tag,
             MPI_Comm comm);
int MPI_Recv(void *buf, int count,
             MPI_Datatype datatype, int source, int tag,
             MPI_Comm comm, MPI_Status status);
```

MPI_Datatype

- MPI_CHAR,
- MPI_INT,
- MPI_UNSIGNED,
- MPI_DOUBLE,
- etc.,
- Custom data types.

MPI_Status & tag

MPI_Status:

- Содержит дополнительную информацию о полученном сообщении.
- MPI_STATUS_IGNORE — специальное значение, которое уменьшает количество используемой памяти в том случае, если вы не собираетесь рассматривать это поле.

tag:

- Число для идентификации сообщения,
- MPI_ANY_TAG.

Блокирующие и не блокирующие операции

MPI_Send/Recv — блокирующиеся функции.



MPI_Isend/Irecv — не блокирующие варианты MPI_Send/Recv.

Вычисление времени

```
double MPI_Wtime()
```

Возвращает время в секундах, прошедшее с некоторого момента в прошлом (точки отсчета). Гарантируется, что эта точка отсчета не будет изменена в течение жизни процесса.

```
double MPI_Wtick()
```

Возвращает разрешение таймера (минимальное значение кванта времени).

```
int MPI_Barrier(MPI_Comm comm)
```

Блокирует работу вызвавшего ее процесса до тех пор, пока все другие процессы группы также не вызовут эту функцию.

Компиляция и запуск

Для компиляции используется `mpicc` вместо `gcc`.

Запуск производится командой вида

```
$ mpirun -np <n_processes> <programm> <args>
```

Спасибо за внимание!

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.