Многопоточные вычисления на основе технологий MPI и OpenMP: Группы и коммуникаторы. Декартова топология

Н. И. Хохлов

МФТИ, Долгопрудный

26 октября 2016 г.

Основные понятие

- Потребность ограничить область коммуникаций некоторым набором процессов, которые составляют подмножество исходного набора.
- Создание коммуникатора на основе подмоножества для общения процессов (коллективные операции).
- MPI предоставляет функционал
  - для работы с группами процессов как упорядоченными множествами;
  - для создания новых коммуникаторов как описателей новых областей связи.

Группа

Группа

Группа – упорядоченное множество процессов.

- в рамках группы каждый процесс имеет целочисленный идентификатор;
- идентификаторы образуют целочисленный ряд, начиная с 0;
- специальный тип данных MPI_Group и набор функций для работы с переменными и константами этого типа.
- Существует две предопределенных группы:
  - MPI_GROUP_EMPTY – группа, не содержащая ни одного процесса;
  - MPI_GROUP_NULL – значение возвращаемое, когда группа не может быть создана.
- Созданная группа не может быть модифицирована (расширена или усечена), может быть только создана новая группа.

Коммуникатор

Коммуникатор

- Скрытый объект с некоторым набором атрибутов, правилами его создания, использования и уничтожения.
- Описывает некоторую область связи.
- Одной и той же области связи может соответствовать несколько коммуникаторов, однако они не являются тождественными и не могут участвовать во взаимном обмене сообщениями.
- Если данные посылаются через один коммуникатор, процесс-получатель может получить их только через тот же самый коммуникатор.
- В MPI существует два типа коммуникаторов:
  - intracommunicator – описывает область связи некоторой группы процессов;
  - intercommunicator – служит для связи между процессами двух различных групп.

MPI_Comm_test_inter

int MPI_Comm_test_inter(MPI_Comm comm, int *flag) – узнать тип коммуникатора.

- comm – коммуникатор;
- flag – возвращает true, если comm – intercommunicator.

При инициализации MPI создается два предопределенных коммуникатора:

- MPI_COMM_WORLD – описывает область связи, содержащую все процессы;
- MPI_COMM_SELF – описывает область связи, состоящую из одного процесса.

Функции для работы с группами

- int MPI_Group_size(MPI_Group group, int *size) – возвращает число процессов в группе.
  - group – группа.
  - size – число процессов в группе.
- Если group = MPI_GROUP_EMPTY, тогда size = 0.
- int MPI_Group_rank(MPI_Group group, int *rank) – возвращает номер процесса в группе.
  - group – группа.
  - rank – номер процесса.
- Если процесс не является членом группы, то возвращается значение MPI_UNDEFINED.

MPI_Group_translate_ranks

int MPI_Group_translate_ranks(MPI_Group group1, int n, int *ranks1, MPI_Group group2, int *ranks2) – функция определяет относительные номера одних и тех же процессов в двух разных группах.

- group1 – группа1;
- n – число процессов, для которых устанавливается соответствие;
- ranks1 – массив номеров процессов из 1-й группы;
- group2 – группа2;
- ranks2 – номера тех же процессов во второй группе.

Если процесс во второй группе отсутствует, то для него устанавливается значениеа MPI_UNDEFINED.

Функции для создания групп

Для создания новых групп в MPI имеется 8 функций.

Группа может быть создана либо с помощью коммуникатора, либо с помощью операций над множествами процессов других групп.

int MPI_Comm_group(MPI_Comm comm, MPI_Group *group) – функция создания группы с помощью коммуникатора.

- comm – коммуникатор;
- group – группа.

Функция создает группу group для множества процессов, входящих в область связи коммуникатора comm.

Создание группы как результат операции над множествами процессов двух групп

- MPI_Group_union(MPI_Group group1, MPI_Group group2, MPI_Group *newgroup);
- MPI_Group_intersection(MPI_Group group1, MPI_Group group2, MPI_Group *newgroup);
- MPI_Group_difference(MPI_Group group1, MPI_Group group2, MPI_Group *newgroup);
  - group1 – группа1;
  - group2 – группа2;
  - newgroup – новая группа.

Создание группы как результат операции над множествами процессов двух групп

- Операции определяются следующим образом:
  - Union – формирует новую группу из элементов 1-й группы и из элементов 2-й группы, не входящих в 1-ю (объединение множеств).
  - Intersection – новая группа формируется из элементов 1-й группы, которые входят также и во 2-ю. Упорядочивание как в 1-й группе (пересечение множеств).
  - Difference – новую группу образуют все элементы 1-й группы, которые не входят во 2-ю. Упорядочивание как в 1-й группе (дополнение множеств).
- Созданная группа может быть пустой, что эквивалентно MPI_GROUP_EMPTY.

MPI_Group_incl, MPI_Group_excl

- MPI_Group_incl(MPI_Group group, int n, int *ranks, MPI_Group *newgroup);
- MPI_Group_excl(MPI_Group group, int n, int *ranks, MPI_Group *newgroup);
  - group – существующая группа;
  - n – число элементов в массиве ranks;
  - ranks – массив номеров процессов;
  - newgroup – новая группа.

MPI_Group_incl, MPI_Group_excl

- Функция MPI_Group_incl создает новую группу, которая состоит из процессов существующей группы, перечисленных в массиве ranks. Процесс с номером i в новой группе есть процесс с номером ranks[i] в существующей группе. Каждый элемент в массиве ranks должен иметь корректный номер в группе group, и среди этих элементов не должно быть совпадающих.
- Функция MPI_Group_excl создает новую группу из тех процессов group, которые не перечислены в массиве ranks. Процессы упорядочиваются как в группе group. Каждый элемент в массиве ranks должен иметь корректный номер в группе group, и среди них не должно быть совпадающих.

MPI_Group_range_incl, MPI_Group_range_excl

- MPI_Group_range_incl(MPI_Group group, int n, int ranges[][3], MPI_Group *newgroup);
- MPI_Group_range_excl(MPI_Group group, int n, int ranges[][3], MPI_Group *newgroup);
  - group – существующая группа;
  - n – число элементов в массиве ranks;
  - ranges – массив областей номеров процессов;
  - newgroup – новая группа.
- Массив rranges представляет собой набор триплетов для задания диапазонов процессов.
- Каждый триплет имеет вид: нижняя граница, верхняя граница, шаг.

Уничтожение группы

- int MPI_Group_free(MPI_Group *group) – уничтожение группы.
  - group – группа.

Функции работы с коммуникаторами

- Разделяются на функции доступа к коммуникаторам и функции создания коммуникаторов.
- Функции доступа являются локальными и не требуют коммуникаций.
- Функции создания, являются коллективными и могут потребовать межпроцессорных коммуникаций.
- Две основных функции доступа к коммуникатору, MPI_Comm_size – опрос числа процессов в области связи и MPI_Comm_rank - опрос идентификатора номера процесса в области связи.

MPI_Comm_compare

Функция сравнения двух коммуникаторов.

int MPI_Comm_compare(MPI_Comm comm1, MPI_Comm comm2, int *result);

- comm1 – первый коммуникатор;
- comm2 – второй коммуникатор;
- result – результат сравнения.

Возможные значения результата сравнения:

- MPI_IDENT – коммуникаторы идентичны, представляют один и тот же объект;
- MPI_CONGRUENT – коммуникаторы конгруэнтны, две области связи с одними и теми же группами;
- MPI_SIMILAR – коммуникаторы подобны, группы содержат одни и те же процессы, но другое упорядочивание;
- MPI_UNEQUAL – во всех других случаях.

MPI_Comm_dup

Функция дублирования коммуникатора. Функция полезна для последующего создания коммуникаторов с новыми атрибутами.

- MPI_Comm_dup(MPI_Comm comm, MPI_Comm *newcomm)
  - comm – коммуникатор;
  - newcomm – копия коммуникатора.

MPI_Comm_create

Функция создания коммуникатора.

- MPI_Comm_create(MPI_Comm comm, MPI_Group group, MPI_Comm *newcomm)
  - comm – коммуникатор;
  - group – группа, для которой создается коммуникатор;
  - newcomm – новый коммуникатор.

Функция создает коммуникатор для группы group. Для процессов, которые не являются членами группы, возвращается значение MPI_COMM_NULL. Функция возвращает код ошибки, если группа group не является подгруппой родительского коммуникатора.

MPI_Comm_split

Функция разделения коммуникатора.

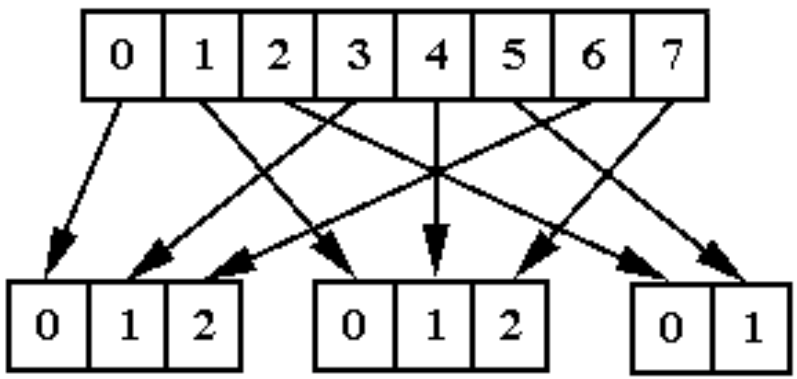
- MPI_Comm_split(MPI_Comm comm, int color, int key, MPI_Comm *newcomm)
  - comm – коммуникатор;
  - color – признак подгруппы;
  - key – управление упорядочиванием;
  - newcomm – новый коммуникатор.

Функция расщепляет группу, связанную с родительским коммуникатором, на непересекающиеся подгруппы по одной на каждое значение признака подгруппы color. Значение color должно быть неотрицательным. Каждая подгруппа содержит процессы с одним и тем же значением color. Параметр key управляет упорядочиванием внутри новых групп: меньшему значению key соответствует меньшее значение идентификатора процесса. В случае равенства параметра key для нескольких процессов упорядочивание выполняется в соответствии с порядком в родительской группе.

Пример

```
MPI_comm comm, newcomm;
int myid, color;
...
MPI_Comm_rank(comm, &myid);
color = myid%3;

MPI_Comm_split(comm, color, myid, &newcomm);
```



Уничтожение коммуникатора

- int MPI_Comm_free(MPI_Comm *comm) – уничтожение коммуникатора.
  - comm – коммуникатор.

Декартова топология

- Обобщением линейной и матричной топологий на произвольное число измерений.
- Для создания коммуникатора с декартовой топологией используется функция MPI_Cart_create.
- Можно создавать топологии с произвольным числом измерений, причем по каждому измерению в отдельности можно накладывать периодические граничные условия.
- Для одномерной топологии мы можем получить или линейную структуру, или кольцо в зависимости от того, какие граничные условия будут наложены.
- Для двумерной топологии, соответственно, либо прямоугольник, либо цилиндр, либо тор.

Создание коммуникатора с декартовой топологией

- MPI_Cart_create(MPI_Comm comm_old, int ndims, int *dims, int *periods, int reorder, MPI_Comm *comm_cart);
  - comm_old – родительский коммуникатор;
  - ndims – число измерений;
  - dims – массив размера ndims, в котором задается число процессов вдоль каждого измерения;
  - periods – логический массив размера ndims для задания граничных условий (true - периодические, false - непериодические);
  - reorder – логическая переменная, указывает, производить перенумерацию процессов (true) или нет (false);
  - comm_cart – новый коммуникатор.

Функция является коллективной, т.е. должна запускаться на всех процессах, входящих в группу коммуникатора comm_old. При этом, если какие-то процессы не попадают в новую группу, то для них возвращается результат MPI_COMM_NULL. Значение параметра reorder=false означает, что идентификаторы всех процессов в новой группе будут такими же, как в старой группе. Если reorder=true, то MPI будет пытаться перенумеровать их с целью оптимизации

Функция определения оптимальной конфигурации сетки

- MPI_Dims_create(int nnodes, int ndims, int *dims);
  - nnodes – общее число узлов в сетке;
  - ndims – число измерений;
  - dims – массив целого типа размерности ndims, в который помещается рекомендуемое число процессов вдоль каждого измерения.

На входе в процедуру в массив dims должны быть занесены целые неотрицательные числа. Если элементу массива dims[i] присвоено положительное число, то для этой размерности вычисление не производится (число процессов вдоль этого направления считается заданным). Вычисляются только те компоненты dims[i], для которых перед обращением к процедуре были присвоены значения 0. Функция стремится создать максимально равномерное распределение процессов вдоль направлений, выстраивая их по убыванию, т.е. для 12-ти процессов она построит трехмерную сетку 4 x 3 x 1. Результат работы этой процедуры может использоваться в качестве входного параметра для процедуры MPI_Cart_create.

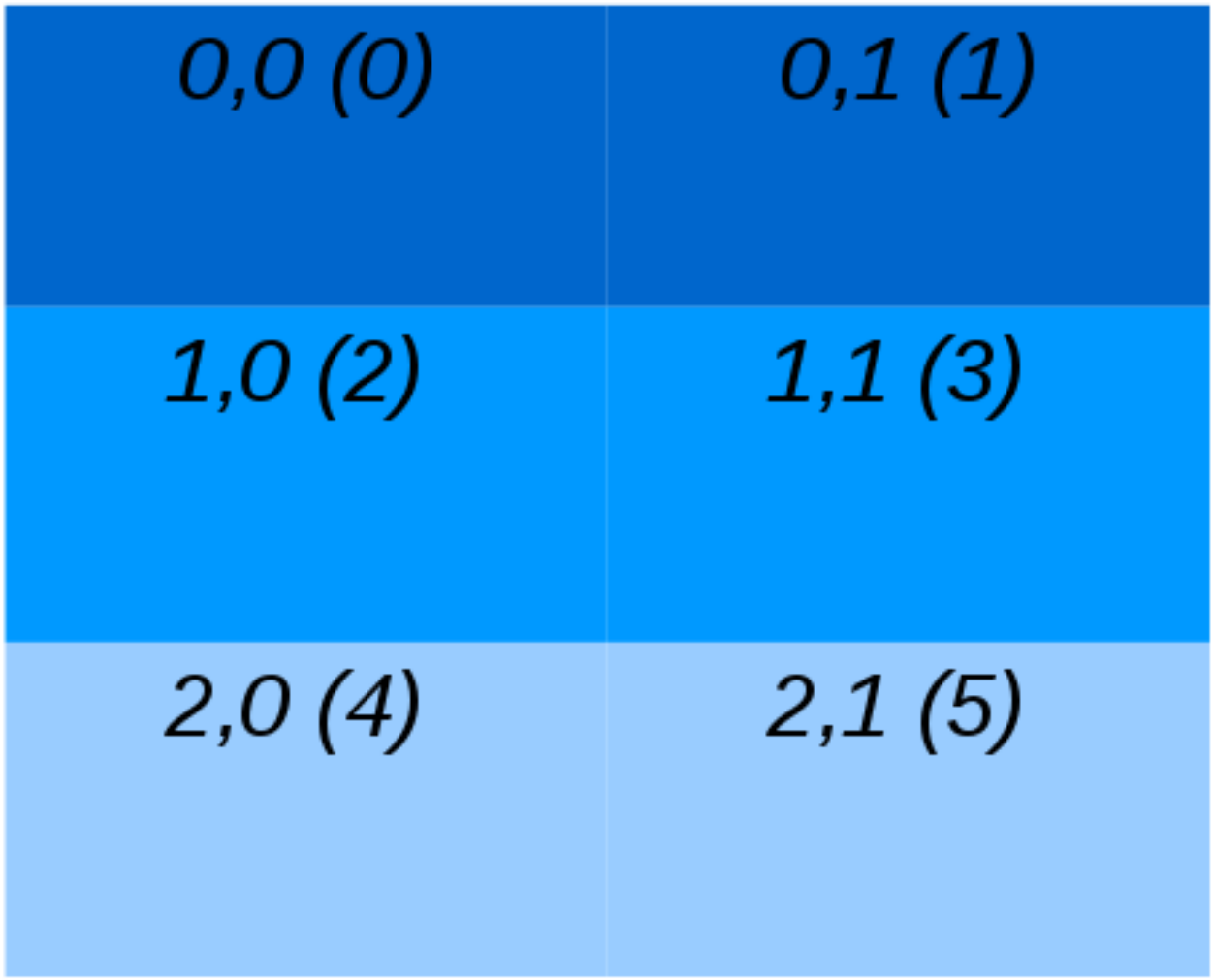
Пример

```
MPI_Comm old_comm, new_comm;
int ndims, reorder, periods[2], dim_size[2];

old_comm = MPI_COMM_WORLD;
ndims = 2;
dim_size[0] = 3;
dim_size[1] = 2;
periods[0] = 1;
periods[1] = 0;
reorder = 1;

MPI_Cart_create(old_comm, ndims, dim_size,
periods, reorder, &new_comm);
```

Пример



Функция опроса числа измерений декартовой топологии

- MPI_Cartdim_get(MPI_Comm comm, int *ndims);
  - comm – коммуникатор с декартовой топологией;
  - ndims – число измерений в декартовой топологии.

Функция возвращает число измерений в декартовой топологии ndims для коммуникатора comm.

MPI_Cart_get

Получение детальной информации о топологии.

- MPI_Cart_get(MPI_Comm comm, int ndims, int *dims, int *periods, int *coords);
  - comm – коммуникатор с декартовой топологией;
  - ndims – число измерений в декартовой топологии;
  - dims – массив размера ndims, в котором возвращается число процессов вдоль каждого измерения;
  - periods – логический массив размера ndims, в котором возвращаются наложенные граничные условия; (true - периодические, false - непериодические);
  - coords – координаты в декартовой сетке вызывающего процесса.

Функция возвращает число измерений в декартовой топологии ndims для коммуникатора comm.

ИНформация о процессах

- MPI_Cart_rank(MPI_Comm comm, int *coords, int *rank) – функция получения идентификатора процесса по его координатам;
  - comm – коммуникатор с декартовой топологией;
  - coords – координаты в декартовой системе;
  - rank – идентификатор процесса.
- MPI_Cart_coords(MPI_Comm comm, int rank, int ndims, int *coords) – функция определения координат процесса по его идентификатору;
  - comm – коммуникатор с декартовой топологией;
  - rank – идентификатор процесса;
  - ndims – число измерений;
  - coords – координаты процесса в декартовой топологии.

Задание. Игра "Жизнь"

- Правила b3/s23.
- Распараллелить используя двумерную декартову топологию.
- Использовать собственные типы данных для пересылок.
- Отметка 2 балла.

Спасибо за внимание! Вопросы?