Руководство пользователя библиотеки коммуникационных процедур INM ParLib (версии 1.0)

Глухов В. Н.

Институт вычислительной математики РАН
Москва, 23 сентября 2020 г.

1 Введение

Данное руководство содержит в себе описание библиотеки коммуникационных процедур INM ParLib, сформировавшейся в результате работ по распараллеливанию моделей погоды [1] и климата [2] на многопроцессорных вычислительных системах с распределенной памятью.

Выделение коммуникационных процедур в отдельную библиотеку было обусловлено возможностью их использования в различных приложениях, в качестве которых могут выступать задачи математической физики, решаемые численно в прямоугольных областях на регулярных сетках, методом методом декомпозиции вычислительных областей по одной или нескольким независимым переменным.

В библиотеке реализованы операция межпроцессорного обмена граничными значениями и операция транспонирования. Первая применяется в случае локальных зависимостей вычислений в некоторой точке сетки от данных в соседних точках. Характерным примером такой зависимости является вычисление производной методом конечных разностей.

Транспонирование применяется тогда, когда существует зависимость от всех точек по одной из распределенных размерностей в то время, как вычисления по другой не распределенной размерности независимы. В этом случае можно перераспределить данные таким образом, что первая размерность будет содержаться целиком в процессорах, а вторая станет распределенной. После завершения вычислений можно вернуться к исходному распределению, сделав обратное транспонирование данных. В некоторых работах [3], [4] была показана эффективность такого подхода для реализации быстрого преобразования Фурье, а также возможность его применения для вычисления преобразования Лежандра [5].

Транспонирование предполагает, что каждый процессор в группе коммутирует с каждым, тогда как при обмене граничными значениями процессоры коммутируют только со своими ближайшими соседями (в данной версии библиотеки ширина коммутируемых границ ограничена размерами локальных подобластей, принадлежащих процессорам).

2 Использование библиотеки

Библиотечные процедуры доступны из программ, написанных как на Си, так и на Фортране.

2.1 Компиляция и линковка программ

Поскольку библиотечные процедуры обращаются к функциям MPI программа должна компилироваться и линковаться точно также как и любая другая программа, использующая библиотеку MPI. Кроме того, при компиляции необходимо указать опцию `-I`*путь_к_библиотеке*`/include`, при линковке – `-L`*путь_к_библиотеке*`/lib -lparlib`, а при линковки программ на Фортране еще и опцию `-lparlibf`.

2.2 Вызов библиотечных процедур

Вызов библиотечных процедур рекомендуется предварять опратором

```
INCLUDE 'parlibf.h'
```

в Фортране, либо директовой

```
#include 'parlib.h'
```

в Си.

2.2.1 Обмен граничными значениями

Пусть массив $\mathcal{A}$ распределен между процессорами по одной или нескольким размерностям. Тогда каждый процессор будет содержать некоторую локальную часть этого массива (подобласть). Предположим, что подобласти не пересекаются между собой. Тогда массив может быть описана в программ как

```
DIMENSION A(STRIDE(1), ..., STRIDE(NDIMS))
```

где `NDIMS` – количество размерностей массива, `STRIDE` – сами размерности. Предположим также, что `BDIM` – одна из размерностей, по которой массив $\mathcal{A}$ распределен и необходимо вдоль нее организовать обмен граничными значениями, при чем в общем случае не для всего массива, а лишь для некоторой его части.

Такой обмен может быть осуществлен путем вызова процедуры `P_BExchange` следующего вида:

```
 CALL  P_BEXCAHNGE (ARR, NDIMS, STRIDE, BLKLEN, BDIM,
1 OVERLAP, DATATYPE, COMM, PERIOD, IERROR)
```

где `ARR` – первый элемент локальной части массива `A`; параметры `BLKLEN` – определяют размеры той части массива `A`, которая участвует в коммуникации, а именно, в коммуникацию будет вовлечена следующая его часть `ARR(1:BLKLEN(1), ..., 1:BLKLEN(NDIMS))`. Ширина коммутируемых границ определяется параметрами `OVERLAP`, при чем предыдущему в группе процессору посылается блок ширины `OVERLAP(1)` по размерности `BDIM`, в то время как следующему – блок ширины `OVERLAP(2)`. Логическая переменная `PERIOD` определяет является ли обмен периодическим или нет. Если обмен периодический, то первый и последний процессоры в группе так же обмениваются между собой граничными значениями. Группа процессор идентефицируется идентификатором MPI `COMM`, а тип элементов массива `A` определяется идентификатором MPI `DATATYPE`. Обычно, последний имеет значения `MPI_REAL` или `MPI_DOUBLE_PRECISION`.

Существует так же асинхронный вариант применения операции обмена граничными значениями. Он реализован в библиотеке в виде четырех процедур:

```
 CALL P_BEXCAHNGE_INIT (NDIMS, STRIDE, BLKLEN, BDIM,
1 OVERLAP, DATATYPE, COMM, PERIO, BEXCHANGE, IERROR)
 CALL P_BEXCAHNGE_START (ARR, BEXCHANGE, IERROR)
 CALL P_BEXCAHNGE_END (BEXCHANGE, IERROR)
 CALL P_BEXCAHNGE_FREE (BEXCHANGE, IERROR)
```

Первая процедура создает и инициализирует некоторую структуру, содержащую в себе всю информацию, необходимую для начала обмена, и возвращает дескриптор `BEXCHANGE`, идентифицирующий эту структуру, вторая начинает обмен, третья ожидает окончания обмена, и, наконец, последняя удаляет созданную структуру из памяти. Таким образом, между процедурами `P_BExcahnge_start` и `P_BExchange_end` могут производиться полезные вычисления в фоновом режиме, одновременно с пересылкой данных.

2.2.2 Транспонирование

Допустим, массив $\mathcal{A}$ распределен между процессорами некоторой группы вдоль размерности DIM_SOURCE блоками ширины LBLKS_SOURCE(IPROC), где IPROC – номер процессора, а размерность DIM_DEST содержится в процессорах целиком. Будем называть транспонированием массива $\mathcal{A}$ такое его перераспределенеие, в результате которого каждый порцессор будет обладать блоком ширины LBLKS_DEST(IPROC) вдоль размерности DIM_DEST и всеми точкамии вдоль размерности DIM_SOURCE. Если ARR_SOURCE – первый элемент локальной подобласти массива $\mathcal{A}$ до транспониравания, а ARR_DEST – после транспонирования, вызов соответствующей библиотечной процедуры будет иметь следующий вид:

```
 CALL P_TRANSPOSE (NDIMS, ARR_SOURCE, DIM_SOURCE,
1 LBLKS_SOURCE, ARR_DEST, DIM_DEST, LBLKS_DEST,
2 STRIDE, BLKLEN, OVERLAP, DATATYPE, COMM, PERIOD,
3 DIAG, IERROR)
```

где NDIMS – количество размерностей массива $\mathcal{A}$, DATATYPE – дескриптор MPI, соответствующий типу элементов массива, COMM – коммуникатор MPI, определяющий группу процессорв. Заметим, что если массив $\mathcal{A}$ хранится в памяти каждого процессора целиком, нет необходимости процессорам пересылать его диагональный блок самим себе. Для того, чтобы пересылки диагонального блока не происходило, переменная DIAG должна иметь значение .FALSE. на входе процедуры.

Пусть локальная подобласть массива $\mathcal{A}$ представлен на входе процедуры массивом A_SOURCE, а на выходе – массивом A_DEST. Они могут быть описаны в программе как

```
DIMENSION A_SOURCE(DIM_SOURCE(1), ..., DIM_SOURCE(NDIMS))
DIMENSION A_DEST(DIM_DEST(1), ..., DIM_DEST(NDIMS))
```

где размерность DIM_SOURCE имеет следующие значения:

$$\texttt{DIM\_SOURSE(IDIM)} = \begin{cases} \texttt{BLKLEN(DIM\_SOURCE)}, \text{ если } \texttt{IDIM=DIM\_SOURCE} \\ \texttt{STRIDE(IDIM)}, \text{ в противном случае} \end{cases}$$

а размерность DIM_DEST:

$$\texttt{DIM\_DEST(IDIM)} = \begin{cases} \texttt{BLKLEN(DIM\_DEST)}, \text{ если } \texttt{IDIM=DIM\_DEST} \\ \texttt{STRIDE(IDIM)}, \text{ в противном случае} \end{cases}$$

В общем счучае массивы A_SOURCE и A_DEST могут быть задействованы в транспонировании не целиком, а лишь частично:

```
ARR_SOURCE(1:BLK_SOURCE(1), ..., 1:BLK_SOURCE(NDIMS)
ARR_DEST(1:BLK_DEST(1), ..., 1:BLK_DEST(NDIMS)
```

где блоки имеют следующие размеры:

$$\texttt{BLK\_SOURCE(IDIM)} = \begin{cases} \texttt{LBLKS\_SOURCE(IPROC)}, \text{ если } \texttt{IDIM=DIM\_SOURCE} \\ \sum \texttt{LBLKS\_DEST}, \text{ если } \texttt{IDIM=DIM\_DEST} \\ \texttt{BLKLEN(IDIM)}, \text{ в противном случае} \end{cases}$$

$$\texttt{BLK\_DEST(IDIM)} = \begin{cases} \texttt{LBLKS\_DEST(IPROC)}, \text{ если } \texttt{IDIM=DIM\_DEST} \\ \sum \texttt{LBLKS\_SOURE}, \text{ если } \texttt{IDIM=DIM\_SOURCE} \\ \texttt{BLKLEN(IDIM)}, \text{ в противном случае} \end{cases}$$

Асинхронный вариант транспонирования реализован в виде четырех процедур:

```
 CALL P_TRANSPOSE_INIT (NDIMS, DIM_SOURCE, LBLKS_SOURCE,
1 DIM_DEST, LBLKS_DEST, STRIDE, BLKLEN, OVERLAP, DATATYPE,
2 COMM, PERIOD, DIAG, TRANSP, IERROR)
 CALL P_TRANSPOSE_START (ARR_SOURCE, ARR_DEST, TRANSP,
1 IERROR)
 CALL P_TRANSPOSE_END (TRANSP, IERROR)
 CALL P_TRANSPOSE_FREE (TRANSP, IERROR)
```

где `TRANSP` – дескриптор транспонирования.

Список литературы

[1] Глухов В. Н. Распараллеливание глобальной спектральной модели среднесрочного прогноза погоды T126 на многопроцессорной вычислительной системе с распределенной памятью. Optimization of Finite Element Approximations, Splines and Wavelets (OFEA'2001). Abstracts of International conference (June 25–29, 2001, St.-Petersburg, Russia), 2001, 184 p.

[2] M. A. Tolstykh, V. N. Gloukhov. Implementation of global atmospheric models on parallel computers. *Вычислительные технологии.* Институт вычислительных технологий СО РАН, Новосибирск.

[3] Ибрагимов И. В. Параллельные алгоритмы БПФ и скалярного произведения векторов. Матричные методы и алгоритмы. ИВМ РАН, Москва, 1999.

[4] P. N. Swarztrauber, S. W. Hammond. A comparison of optimal FFTs on torus and hypercube multicomputers. *Parallel computing*, 27 (2001), pp. 847–859.

[5] Barros, S. R. M., Kauranne, T. On the parallelization of global spectral weather models. *Parallel Computing*, 20 (1994), pp. 1335–1356.

[6] B. Rodriguez, L. Hart, T. Henderson. Performance and portability in parallel computing: a weather forecasr view. *High Performance Computing in the Geosciences*, 1995, Kluwer Academic Publishers, Netherlands, pp. 1–23.

[7] T. Henderson, D. Shaffer, M. Govett, L. Hart. SMS User's Guide. Advanced Computing Branch, Aviation Division, NOAA Forecast system laboratory, Boulder, 2001.