

ЭЛЕКТРОННЫЙ ЗАДАЧНИК ПО ПАРАЛЛЕЛЬНОМУ MPI-ПРОГРАММИРОВАНИЮ ДЛЯ ЯЗЫКОВ C++ И PASCAL С ВОЗМОЖНОСТЬЮ РАСШИРЕНИЯ БАЗОВОГО НАБОРА ЗАДАНИЙ

М.Э. Абрамян

В статье рассматривается один из подходов к разработке электронных задачник по параллельному MPI-программированию и реализованный на основе этого подхода программный комплекс Programming Taskbook for MPI. Приводятся примеры учебных заданий и описывается конструктор, позволяющий разрабатывать новые группы заданий.

Об одном из подходов к разработке электронных задачник по параллельному программированию

Общеизвестно, какую важную роль при изучении программирования играют практические занятия. Уверенное владение различными алгоритмами и связанными с ними средствами того или иного языка программирования может быть достигнуто лишь после выполнения большого числа практических заданий, связанных с разработкой учебных программ.

Наличие специальных программных средств, упрощающих разработку учебных программ, позволило бы сделать процесс практического изучения программирования более быстрым и эффективным. Комплекс подобных программных средств (который естественно назвать *электронным задачником* [1]) мог бы автоматически создавать проекты-заготовки для решения поставленного задания, предоставлять исходные данные для тестирования созданной программы и обеспечивать автоматическую проверку правильности предложенного решения. Перечисленные возможности оказываются полезными не только при изучении «обычного», непараллельного программирования, но и при знакомстве с методами и средствами параллельного программирования, основанного на передаче сообщений (технология MPI). При выполнении заданий по параллельному MPI-программированию электронный задачник дополнительно мог бы автоматически запускать требуемое число параллельных процессов, предоставлять каждому из этих процессов свой набор исходных данных и получать от каждого процесса результирующие данные для проверки их правильности и вывода на экран. Особенно полезным было бы использование такого задачника на начальных этапах изучения параллельного программирования.

Поскольку программа разрабатывается в конкретной среде, целесообразно интегрировать задачник в изучаемую программную среду. Однако принцип интеграции задачника и программной среды не означает, что для каждой среды надо разрабатывать свой вариант задачника. Используя современные средства программирования, можно создать универсальное «ядро» задачника и набор модулей-«посредников», с помощью которых это ядро будет интегрировано в различные программные среды, в том числе и ориентированные на разные языки программирования. В операционной системе Windows «ядро» задачника естественно реализовать в виде динамически подключаемой библиотеки (dll), поскольку средства подключения таких библиотек к программе предусмотрены в подавляющем большинстве программных сред, функционирующих под управлением Windows. Помимо процедур, управляющих работой задачника, «ядро» должно содержать достаточно обширный базовый набор заданий по различным разделам программирования. Весьма желательным представляется также возможность расширения задачника, т. е. дополнения имеющегося базового набора новыми группами заданий. Новые группы можно было бы также оформлять в виде динамических библиотек, автоматически подключаемых к ядру задачника при его инициализации.

Основные возможности электронного задачника Programming Taskbook for MPI

Перечисленные выше принципы реализации электронного задачника были положены автором в основу разработанного им программного комплекса Programming Taskbook [2] — универсального электронного задачника по программированию, доступного для использования в различных программных средах. На базе этого задачника было разработано его дополнение — комплекс «Электронный задачник по параллельному MPI-программированию Programming Taskbook for MPI» (PT for MPI). Данный комплекс содержит дополнительные компоненты задачника Programming Taskbook, которые позволяют выполнять задания на разработку параллельных программ с применением технологии MPI.

Задания по параллельному MPI-программированию с использованием задачника PT for MPI могут выполняться на языках Pascal и C++ в следующих программных средах:

- Borland Delphi 7.0 и Turbo Delphi 2006;
- Free Pascal Lazarus 0.9 [3];
- PascalABC.NET [4];
- Microsoft Visual C++ 6.0;
- Microsoft Visual Studio .NET 2003, 2005 и 2008.

Помимо одной из перечисленных программных сред на компьютере должна быть также установлена система MPICH для Windows версии 1.2.5 [5], используемая для запуска программ учащихся в параллельном режиме.

К программам учащегося должна подключаться динамическая библиотека mpich.dll, входящая в состав комплекса MPICH. Для программ на языке C++ доступ к библиотеке обеспечивается с помощью файла mpich.lib и набора заголовочных файлов (также входящих в комплекс MPICH), а для программ на языке Pascal — с помощью модуля MPI.pas, который разработан автором задачника PT for MPI и содержит описания констант, типов и более 120 функций MPI стандарта MPI-1.1. Включение языка Pascal в задачник по параллельному программированию связано с тем, что этот язык активно используется в учебных целях и является типичным представителем императивных языков с возможностью прямого доступа к динамической памяти посредством указателей, т. е. относится к той категории языков программирования, на которую и ориентирована технология MPI. Таким образом, знания, полученные при изучении MPI с применением языка Pascal, можно в дальнейшем использовать и при разработке параллельных программ на других языках.

Задачник PT for MPI предоставляет при выполнении заданий те же возможности, что и базовый задачник Programming Taskbook; в частности, он передает программе учащегося исходные данные, проверяет правильность результатов, полученных программой, и сохраняет сведения о каждом тестовом испытании программы в специальном файле результатов. Кроме того, в задачнике PT for MPI предусмотрены дополнительные возможности, связанные со спецификой выполнения заданий по параллельному программированию:

- демонстрационный просмотр заданий, не требующий использования параллельного режима;
- создание для выбранного задания проекта-заготовки с подключенными к нему модулями библиотеки MPI;
- особый механизм, обеспечивающий выполнение программы учащегося в параллельном режиме при ее обычном запуске из среды разработки: запущенная программа выполняет запуск приложения MPIRun.exe из комплекса MPICH, которое, в свою очередь, запускает программу в параллельном режиме (все процессы выполняются на локальном компьютере);
- передача каждому процессу параллельной программы его собственного набора исходных данных;
- получение от каждого процесса требуемых результатов и их автоматическая пересылка в главный процесс для проверки и отображения в окне задачника;
- вывод информации об ошибках времени выполнения (в том числе ошибках, возникших при выполнении функций MPI) и ошибках ввода-вывода с указанием рангов процессов, в которых эти ошибки произошли;
- возможность вывода отладочной информации для каждого процесса в специальном разделе окна задачника;
- автоматическая выгрузка из памяти всех запущенных процессов даже в случае зависания параллельной программы.

Перечисленные выше возможности избавляют учащегося от необходимости выполнения дополнительных действий, связанных с запуском его программы в параллельном режиме, и упрощают выявление и исправление стандартных ошибок, возникающих в параллельных программах.

Использование исходных данных, подготовленных задачником для каждого процесса параллельной программы, наглядный вывод в одном окне всех результатов, полученных каждым процессом, и их автоматическая проверка, а также дополнительные средства отладки позволяют учащемуся сосредоточиться на реализации алгоритма решения задачи и обеспечивают надежное тестирование разработанного алгоритма.

Базовый набор заданий и особенности выполнения учебных заданий с использованием электронного задачника

В состав задачника PT for MPI включена группа учебных заданий MPIBegin, которая содержит 100 заданий, предназначенных для освоения библиотеки MPI. Задания разбиты на следующие подгруппы (в скобках указано количество заданий в каждой подгруппе):

- «Процессы и их ранги» (6),
- «Обмен сообщениями между отдельными процессами» (23),
- «Коллективная пересылка данных» (18),
- «Коллективные операции редукции» (10),
- «Производные типы и упаковка данных» (13),
- «Группы процессов и коммутаторы» (12),
- «Виртуальные топологии» (18).

В качестве примера приведем формулировку достаточно простого задания из подгруппы «Обмен сообщениями между отдельными процессами».

MPIBegin17. В каждом процессе дано вещественное число. Переслать число из главного процесса во все подчиненные процессы, а все числа из подчиненных процессов — в главный, и вывести в каждом процессе полученные числа (в главном процессе числа выводить в порядке возрастания рангов переславших их процессов).

Приступая к выполнению задания, учащийся может выбрать требуемую среду программирования из числа имеющихся на данном компьютере и создать в ней проект-заготовку для нужного задания, используя специальный программный модуль PT4Load, входящий в состав задачника Programming Taskbook. Если проект предназначен для выполнения задания по параллельному программированию, то к нему автоматически подключаются модули с реализацией библиотеки MPI, а в его код добавляются операторы, необходимые практически в любой параллельной программе. Приведем начальную часть кода для языка C++, созданного автоматически при выборе задания MPIBegin17 (среда программирования Microsoft Visual C++ 6 или Microsoft Visual Studio .NET 2003, 2005, 2008):

```
#include <windows.h>
#pragma hdrstop
#include "pt4.h"
#include "mpi.h"

void Solve()
{
    Task("MPIBegin17");
    int flag;
    MPI_Initialized(&flag);
    if (flag == 0)
        return;
    int rank, size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
}
```

В данном фрагменте указываются необходимые заголовочные файлы и определяется функция Solve, в которой требуется запрограммировать решение задачи. Первым оператором функции Solve является вызов функции Task, инициализирующей выбранное задание. Прочие операторы позволяют определить, запущено ли задание в параллельном режиме, а также узнать число процессов параллельного приложения и ранг текущего процесса (вызов функции MPI_Init, инициализирующей параллельный режим, отсутствует, так как данная функция автоматически вызывается при инициализации любого задания по параллельному программированию).

Для выполнения задания MPIBegin17 в функцию Solve достаточно добавить следующий фрагмент кода (можно также воспользоваться комбинированной функцией MPI_Sendrecv):

```
double a;
MPI_Status s;
pt >> a;
if (rank > 0)
{
    MPI_Send(&a, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
    MPI_Recv(&a, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD, &s);
    pt << a;
}
else
    for (int i = 1; i < size; ++i)
    {
        double b;
        MPI_Recv(&b, 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD, &s);
        pt << b;
        MPI_Send(&a, 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD);
    }
```

Операторы ввода-вывода, использующие специальный поток pt, позволяют программе учащегося получать исходные данные, подготовленные задачиком, и передавать задачику результаты для их проверки и отображения на экране. Таким образом, благодаря применению задачника существенно упрощаются те этапы выполнения задания, которые не связаны с собственно разработкой алгоритма, а именно ввод исходных данных и вывод результатов. Заметим, что при дальнейшем изучении библиотеки MPI можно вернуться к этому заданию и выполнить его другим способом, используя функции коллективной пересылки данных.

Для ускорения тестирования и отладки программ, выполняющих задания по параллельному программированию, в задачнике PT for MPI реализован специальный механизм, позволяющий запустить

параллельное приложение непосредственно из интегрированной программной среды. Программа, запущенная из интегрированной среды, выступает в роли «загрузчика» параллельного приложения, причем все ее действия по запуску параллельного приложения осуществляются автоматически. Основные этапы выполнения программы, решающей параллельную задачу, приведены на рис. 1.



Рис. 1.

Результаты каждого тестового запуска программы немедленно отображаются на экране (см. рис. 2, на котором приводится окно задачника при правильном выполнении задания MPIBegin17). При этом учащийся может ознакомиться с предложенным набором исходных данных, результирующими данными, полученными каждым процессом его параллельной программы, а также контрольным («правильным») набором результатов. Задачник автоматически проверяет правильность операций ввода-вывода и при выполнении ошибочных действий (например, при попытке ввести лишние исходные данные или использовать данные неверного типа) выводит соответствующие сообщения. Помимо вывода на экран результаты каждого запуска программы фиксируются в специальном зашифрованном файле результатов, просмотреть который можно с помощью программного модуля PT4Results, входящего в состав задачника.

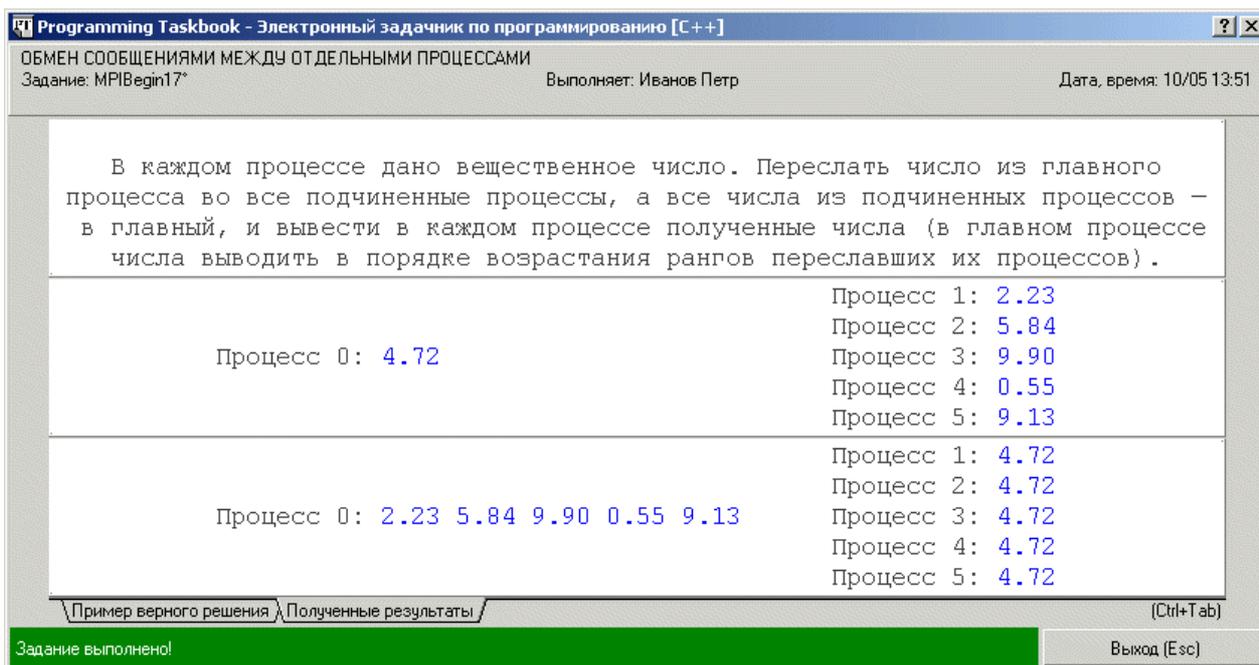


Рис.2.

Приведем формулировки еще двух заданий из группы MPIBegin. Задание MPIBegin73 относится к подгруппе «Группы процессов и коммутаторы», а задание MPIBegin87 — к подгруппе «Виртуальные топологии».

MPIBegin73. В каждом процессе, ранг которого делится на 3 (включая главный процесс), даны три целых числа. С помощью функции `MPI_Comm_split` создать новый коммутатор, включающий процессы, ранг которых делится на 3. Используя одну коллективную операцию пересылки данных для созданного коммутатора, переслать исходные числа в главный процесс и вывести эти числа в порядке возрастания рангов переславших их процессов (включая числа, полученные из главного процесса).

MPIBegin87. Число процессов K кратно трем: $K = 3N$, $N > 1$. В процессах 0, N и $2N$ дано по N целых чисел. Определить для всех процессов декартову топологию в виде матрицы размера $3 \times N$, после чего, используя функцию `MPI_Cart_sub`, расщепить матрицу процессов на три одномерные строки (при этом процессы 0, N и $2N$ будут главными процессами в полученных строках). Используя одну коллективную операцию пересылки данных, переслать по одному исходному числу из главного процесса каждой строки во все процессы этой же строки и вывести полученное число в каждом процессе (включая процессы 0, N и $2N$).

Для отладки параллельных программ задачник предоставляет дополнительные средства, обеспечивающие вывод информации в специальный раздел отладки окна задачника (в этом же разделе выводятся сообщения об ошибках, которые произошли в различных процессах). В разделе отладки можно просматривать информацию, выведенную каким-либо процессом параллельной программы, а также сводную информацию по всем процессам. При просмотре можно скрыть все разделы окна задачника, кроме раздела отладки (см. рис. 3).

```
0| 1> Ранг процесса: 0
0| 2> Элемент исходных данных: 37
1| 1> Ранг процесса: 1
1| 2> Элемент исходных данных: 60
1| 3>
1| 4> !Выведены не все результирующие данные.
1| 5> ! Количество выведенных данных: 0 (из 1).
2| 1> Ранг процесса: 2
2| 2> Элемент исходных данных: 23
2| 3>
2| 4> !Выведены не все результирующие данные.
2| 5> ! Количество выведенных данных: 0 (из 1).
3| 1> Ранг процесса: 3
3| 2> Элемент исходных данных: 79
3| 3>
3| 4> !Выведены не все результирующие данные.
3| 5> ! Количество выведенных данных: 0 (из 1).
```

Рис.3

Реализованные в задачнике PT for MPI средства автоматического запуска и отладки параллельных приложений позволяют использовать его для разработки и тестирования параллельных программ, не связанных с конкретными учебными задачами. С этой целью в задачник PT for MPI включена вспомогательная группа MPIDebug из 36 «заданий», каждое из которых обеспечивает автоматический запуск параллельной программы, причем количество процессов определяется порядковым номером задания. Таким образом, эти «задания» позволяют запускать любые параллельные программы с требуемым количеством процессов непосредственно из интегрированной среды и предоставляют для их отладки средства, имеющиеся в задачнике.

Средства расширения базового набора заданий

С помощью конструктора учебных заданий `PT4TaskMaker` [6], входящего в состав задачника `Programming Taskbook`, преподаватель может разрабатывать дополнительные группы заданий, в том числе связанные с параллельным MPI-программированием. Текущая версия конструктора позволяет разрабатывать задания на языке Pascal в средах Delphi и Lazarus, причем разработанные задания будут доступны для выполнения в любой программной среде, в которую интегрирован задачник. При реализации каждого задания необходимо указать его формулировку, сгенерировать (с использованием датчика случайных чисел) набор исходных данных и определить набор контрольных, т. е. «правильных», результирующих данных. В заданиях по параллельному MPI-программированию дополнительно требуется определить количество процессов, которые будут запущены при выполнении задания (это количество также рекомендуется определять случайным образом из некоторого допустимого диапазона, для того чтобы алгоритм решения задачи был протестирован более надежно). Кроме того, в заданиях по параллельному программированию необходимо для каждого элемента

исходных (или результирующих) данных указывать ранг процесса, который должен получить от задачника (или, соответственно, передать задачнику) этот элемент данных.

Созданные группы заданий оформляются в виде динамических библиотек (dll-файлов). Для подключения созданной группы к задачнику достаточно разместить dll-файл с ее реализацией в подкаталоге LIB системного каталога задачника или в рабочем каталоге учащегося. После этого все задания новой группы станут доступными для выполнения, а информация об этой группе будет включена в программные модули задачника PT4Demo и PT4Load, обеспечивающие соответственно просмотр заданий в демонстрационном режиме и создание программ-заготовок для выполнения заданий в различных программных средах.

В задачнике Programming Taskbook и его расширении PT for MPI предусмотрена возможность автоматической генерации html-страниц с описанием любой группы заданий (как входящей в базовый набор, так и разработанной с помощью конструктора учебных заданий). Помимо текста формулировок всех заданий и заголовка группы страница может содержать заголовки подгрупп в пределах данной группы, а также комментарии ко всей группе заданий и ее подгруппам. Для генерации html-страницы и ее немедленного отображения в стандартном веб-браузере можно использовать программный модуль PT4Demo, входящий в состав задачника; кроме того, сгенерировать требуемую страницу можно и непосредственно из программы, выполняющей учебное задание, вызвав соответствующую процедуру. Эта возможность задачника делает его «самодокументируемым» и упрощает ознакомление с имеющимися группами заданий.

Подробная информация о возможностях задачника PT for MPI содержится на сайте ptaskbook.com. Там же приводятся формулировки групп заданий MPIDebug и MPIBegin, решения типовых задач и примеры реализации новых заданий с использованием конструктора PT4TaskMaker. С этого сайта можно загрузить последние версии как задачника Programming Taskbook, так и его расширения PT for MPI.

ЛИТЕРАТУРА:

1. М.Э. Абрамян. Реализация универсального электронного задачника по программированию // Информатика и образование, 2009, № 6. С. 118–120.
2. М.Э. Абрамян. Электронный задачник по программированию для языков Pascal, C++, Visual Basic, VB.NET, C# с возможностью расширения базового набора заданий / VII Международная конференция памяти акад. А. П. Ершова «Перспективы систем информатики», Новосибирск, 15–19 июня 2009 г. Секция «Информатика образования». Доклады и тезисы. Новосибирск, 2009. С. 14–19.
3. Free Pascal Lazarus 0.9, <http://sourceforge.net/projects/lazarus/>.
4. PascalABC.NET, <http://pascalabc.net/>.
5. MPICH for Win32 1.2.5, <ftp://ftp.mcs.anl.gov/pub/mpi/nt/mpich.nt.1.2.5.exe>.
6. М.Э. Абрамян. Новые возможности задачника Programming Taskbook: обработка динамических структур и конструктор учебных заданий / IV Международная научно-практическая конференция «Современные информационные технологии и ИТ-образование», Москва, 14–15 декабря 2009 г. Сборник трудов. М.: ИНТУИТ.РУ, 2009. С. 282–289.