

Лабораторные задания по курсу “Технологии и компьютерные системы обработки данных” (MPI+OpenMP)

Задание Л1 (5 баллов).

В рамках MPI реализовать механизм сборки данных в соответствии со следующими положениями:

- 1) Каждый процесс владеет некоторым собственным набором данных одного и того же типа. Данные хранятся в памяти процесса в виде линейного массива (это может быть символьный массив с сообщением типа “Привет от процесса номер такой-то”).
- 2) Необходимо собрать наборы данных со всех процессов на нулевой процесс с сохранением их на нем в виде двумерного динамического массива: данные i -го процесса помещаются в i -ю строку массива.

Важно!

- 1) Организацию сборки данных необходимо выполнить, исходя из предположения о том, что количество данных на каждом процессе может быть различным. При этом 0-процесс для получения и размещения данных от i -го процесса должен выделить ровно столько памяти (под i -ю строку массива), сколько для этого необходимо.
- 2) С целью уменьшения времени простоя данные от процессов необходимо собирать по мере их готовности (поступления), а не в порядке нумерации процессов.

Задание Л2 (20 баллов).

Разработать MPI-приложение для расчета интеграла методом трапеции на заданном количестве процессов. Интервал интегрирования разбивается на n отрезков, затем множество отрезков последовательно разбивается на блоки отрезков величины r , количество блоков должно быть больше количества процессов p . Далее необходимо реализовать динамическое распределение вычислительной нагрузки в рамках “master-slave” модели: мастер-процесс (0-процесс) по запросу распределяет по одному блоку между ненулевыми процессами (на старте можно без запроса), рабочий процесс получает задание-блок, рассчитывает частичную сумму и делает запрос на следующий блок. Частичные суммы процесс может накапливать локально, передавая результат мастер-процессу в конце обработки своего набора блоков.

Примечания:

Входные параметры задачи: количество процессов p , количество отрезков разбиения интервала интегрирования n , размер блока отрезков r для динамического распределения вычислительной нагрузки.

Задание Л3 (20 баллов).

Реализовать параллельный алгоритм решения задачи, последовательный код которой имеет вид:

```
int a[N1][N2];
```

```

for (i = 1; i < N1; i++)
    for (j = 0; j < N2; j++)
        a[i][j] = a[i-1][j] + 1;

```

В рамках этого кода происходит последовательная инициализация элементов двумерного массива: элементы i -строки инициализируются соответствующими (стоящими в том же столбце) элементами $(i-1)$ -строки, увеличенными на единицу.

С параллельной точки зрения, двумерный массив нарезается на ленты строк по количеству процессов. Каждая лента содержит $N1/P$ строк. Ленты распределяются между процессами: расчет строк i -ленты осуществляется на i -процессе. Значения первой строки ленты на процессе зависят от значений последней строки ленты соседнего процесса с меньшим номером. Данные зависимости порождают необходимость в обмене данными между процессами.

С целью оптимизации коммуникаций ленты строк нарезаются по столбцам на блоки. Всего $N2/g$ блоков, где g – размер блока. В рамках параллельного алгоритма, каждый MPI-процесс последовательно блок за блоком инициализирует их элементы. Расчет блоков перемежается операциями обмена данными. Перед расчетом элементов i -блока процесс должен получить элементы последней строки i -блока от соседнего процесса с меньшим номером, затем с помощью полученных значений выполнить инициализацию элементов своего i -блока и, после этого, передать последнюю строку этого блока соседнему процессу с большим номером.

Важно!

Вычислительная нагрузка в данном алгоритме на самом деле не принципиальна и выбрана просто для конкретики. Главное в этом задании – оптимальная организация обменов данными. В идеале, все коммуникации необходимо спрятать под вычисления (overlapping communication and computation).

Задание Л4 (35 баллов).

Разработать MPI-программу для умножения матриц по блочному алгоритму Фокса.

Задание Л5 (10+20 баллов).

В рамках технологии OpenMP разработать параллельное приложение для численного решения двумерного уравнения эллиптического типа (задача Дирихле)

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = f(x_1, x_2, t), \quad (x_1, x_2) \in D = [0; l_1] \times [0; l_2],$$

$$u|_{\partial D} = \mu(x_1, x_2)$$

- 1) по схеме чередования четных и нечетных строк,
- 2) методом Гаусса-Зейделя

Дополнительно (по желанию)

Задание Л6 (*).

Разработать пользовательскую версию функции MPI_Bcast() с сохранением оригинального списка аргументов.

Задание Л7 (*).

В рамках технологии OpenMP реализовать параллельный алгоритм линейной прогонки для решения систем уравнений с трехдиагональной матрицей.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Антонов А.С. Технологии параллельного программирования MPI и OpenMP. М.: Изд-во МГУ, 2012. 344 с.
2. Таненбаум Э., Остин Т. Архитектура компьютера. СПб.: Питер, 2015. 816 с.
3. Гегель В.П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем. М.: Изд-во МГУ, 2010. 544 с.
4. White T. Hadoop: The Definitive Guide, O'Reilly Media, Inc., 2015. 728 p.

Перечень дополнительной литературы

1. Гегель В.П. Теория и практика параллельных вычислений. Москва: Бином, 2007. 423 с.
2. Таненбаум Э., Уэзеролл Д. Компьютерные сети. СПб.: Питер, 2016. 960 с.
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002г. 608 с.