

Факультет компьютерных наук

Кафедра вычислительных систем

КОНТРОЛЬНЫЕ ЗАДАНИЯ

По дисциплине «Параллельные вычисления»

Тема: «Ускорение вычисления числовых рядов с использованием кластера»

Цель работы: Выработка и закрепление навыков:

- ✓ разработки параллельных программ для вычисления значений медленно сходящихся числовых рядов;
- ✓ распределения вычислительной работы между процессорами;
- ✓ распределения данных между процессорами;
- ✓ обмена необходимыми данными.

Задание:

Разработать параллельную программу для вычисления значения числового ряда с помощью нескольких рабочих станций кластера.

Указания к выполнению

1. Получить вариант задания на поиск значения числового ряда.
2. Для хранения файлов создайте папку, назвав ее фамилией студента
3. Напишите текст последовательной программы.
4. Включите в программу функции обращения к таймеру для определения времени вычисления ряда. Предусмотрите вывод числа запущенных процессов, значения ряда, погрешности и времени вычисления в окно консоли. В качестве погрешности используется разница между известным значением результата, например π или $\ln(2)$, и вычисленным (если это возможно).
5. Количество элементов ряда должно задаваться путем ввода по запросу программы или из командной строки.
6. Предусмотрите вызов функций определения текущего времени перед вычислением ряда и после вычисления. Определите время вычисления значения ряда как разницу данных величин. Перед функциями определения текущего времени выполните синхронизацию всех запущенных процессов. Время на выполнение ввода и вывода из программы не включайте в искомую величину.
7. Наберите текст в среде программы Microsoft Visual Studio. После отладки выполните последовательную версию программы. Задайте число элементов таким, чтобы время выполнения программы было более 1 сек.
8. Разработайте параллельную программу вычисления ряда, предусмотрев возможность ее выполнения на произвольном количестве рабочих станций кластера.
9. Распределите вычислительную работу и данные между процессорами в предположении, что они имеют одинаковую производительность. При распараллеливании распределите слагаемые между процессорами с некоторым шагом индекса элемента ряда, кратным числу процессоров.
10. Для знакопеременных рядов слагаемые следует распределять по процессорам парами (положительное с отрицательным). В противном случае будут накапливаться очень большие значения с разными знаками, что приведет к потере точности результата.
11. В конце сложите все частичные суммы в нулевом узле. Предусмотрите вывод результата только от нулевого процесса.
12. Создайте новый проект. Выполните все настройки среды выполнения программы для параллельной реализации, как в первой лабораторной работе. В табл.1 содержится напоминание настроек среды.

Таблица Настройки среды программы Visual Studio 2008 (2005)

Configurati on	Ветвь из Configu-ration Properties	Подветвь	Имя параметра	Значение параметра
Debug	C/C++	Code Generation	Runtime Library	Debug Multithreaded.../MTd/...
	Linker	General	Additional Directories	C:\Programs\Mpich\SDK\Lib
		Debugging	Generate Debug Info	Yes (/DEBUG)
		Input	Additional Dependencies	ws2_32.lib mpichd.lib
Release	C/C++	Code Generation	Runtime Library	Multi-threaded (/MT)
	Linker	Input	Additional Dependencies	ws2_32.lib mpich.lib
All Configuratio ns	C/C++	General	Additional Include Directories	C:\Programs\Mpich\SDK\Inc lude
	Linker	General	Additional Directories	C:\Programs\Mpich\SDK\Lib
		Input	Additional Dependencies	ws2_32.lib mpich.lib

При выполнении настроек следует иметь в виду, что в связи с постоянным обновлением среды Visual Studio в настройках могут происходить некоторые изменения.

13. Наберите текст параллельной программы. Отладьте и выполните программу сначала в виде одного процесса прямо в среде Microsoft Visual Studio.
14. Скопируйте выполняемый файл в папку D:\MpichRun каждой из станций кластера.
15. Иницилируйте работу базовой системы запуска MPIRun.
16. Запустите на выполнение параллельную программу сначала в виде одного процесса на своем компьютере. Не забывайте задать число элементов, чтобы время выполнения было порядка нескольких секунд. Число элементов ряда желательно задать в виде 100...0 (единица с некоторым числом нулей): обычно от 1 млн. до 1 млрд.
17. При определении времени выполняйте неоднократный запуск программы. Правильным будет минимальное время из нескольких попыток. Неправильный результат будет в том случае, если узел, на котором выполняется ваша ветвь программы, занят другой работой.
18. Выполните программу на 2, 4, 8 процессорах. Зафиксируйте все результаты. Убедитесь, что при увеличении числа процессов в два раза время выполнения программы уменьшается тоже примерно в два раза.
19. С помощью диспетчера задач убедитесь, что программа выполняется не только на вашем компьютере. Для этого один раз нажмите Ctrl/Alt/Del и на вкладке «Процессы», отсортировав колонку «Имя образа» путем щелчка на имени колонки, найдите свой процесс. Не забывайте, что время нахождения процесса в этом списке равно времени выполнения программы, то есть нескольким секундам. За это время нужно успеть посмотреть список процессов.
20. Результат оформить в виде отчета на бумажном носителе информации. Кроме задания и текста программы включить в него результаты выполнения программы.

Варианты заданий (24 варианта):

$$R1 = 1 + \sum_{n=1}^{\infty} \frac{1}{\sqrt{n}}$$

$$R2 = \sum_{n=1}^{\infty} \frac{1}{n^2}$$

$$R3 = \sum_{n=1}^{\infty} \frac{1}{(2n-1)^2}$$

Простейший гармонический ряд

$$R4 = \sum_{n=1}^{\infty} \frac{1}{n}$$

Гармонические ряды

$$R5 = \sum_{n=1}^{\infty} \frac{1}{n^{\alpha}} \quad \alpha > 1$$

$$R6 = \sum_{n=1}^{\infty} \frac{1}{\sqrt{(n(n^2+1))}}$$

$$R7 = \sum_{n=2}^{\infty} \frac{1}{(\ln n)^{\ln n}}$$

$$R8 = \sum_{n=1}^{\infty} \left(\frac{1}{n} - \ln \frac{n+1}{n} \right)$$

Известные константы

$$R9 = \pi = \sqrt{6 \sum_{n=1}^{\infty} \frac{1}{n^2}}$$

$$R10 = \pi = \sqrt[4]{90 \sum_{n=1}^{\infty} \frac{1}{n^4}}$$

$$R11 = \pi = \frac{6}{\sqrt{3}} \sum_{n=0}^{\infty} \frac{(-1)^{n+2}}{(2n+1)3^n}$$

$$R12 = \pi = \sqrt{12 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^2}}$$

Ряд Лейбница

$$R13 = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1}$$

Знакопеременные (логарифмические) ряды

$$R14 = \ln 2 = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}$$

$$R15 = \ln(1+x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n \quad |x| < 1$$

$$R16 = \ln 2 = \sum_{n=1}^{\infty} \left(\frac{1}{2n-1} - \frac{1}{2n} \right)$$

$$R17 = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{(2n-1)^2}$$

$$R18 = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^2}$$

$$R19 = \ln 2 = \sum_{n=1}^{\infty} \frac{1}{(2n-1)2n}$$

$$R20 = \ln 5 = \frac{4}{3} \sum_{n=0}^{\infty} \frac{1}{(2n+1)9^n} + \frac{2}{9} \sum_{n=0}^{\infty} \frac{1}{(2n+1)81^n}$$

$$R21 = \ln 2 = \frac{2}{3} \sum_{n=0}^{\infty} \frac{1}{(2n+1)9^n}$$

$$R22 = \sum_{n=1}^{\infty} \frac{x^n}{n^2} \quad |x| < 1$$

Показательная и тригонометрическая функции

$$R23 = e^x = \sum_{n=1}^{\infty} \left(1 + \frac{x}{n}\right)^n$$

$$R24 = \operatorname{arctg}(x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1} x^{2n-1} \quad |x| < 1$$

**Пример программы на вычисление ряда
(определение числа π)**

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
double f(double a)
{
    return (4.0 / (1.0 + a*a));
}

void main (int argc, char *argv[])
```

$$R = \frac{1}{n} \sum_{i=1}^{\infty} \frac{4}{1 + \left(\frac{i-0.5}{n}\right)^2}$$

```

{
int done = 0, n, myid, numprocs, i;
double PI25DT = 3.141592653589793238462643;
double mypi, pi, h, sum, x;
double startwtime, endwtime;
int namelen;
char processor_name[MPI_MAX_PROCESSOR_NAME];

MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);
MPI_Get_processor_name(processor_name,&namelen);
printf("Process %d on %s\n", myid, processor_name);
fflush(stdout);

n = 0;
while (!done)
{
if (myid == 0)
{
printf("Enter the number of intervals: (0: exit) ");
fflush(stdout);
scanf("%d",&n);
startwtime = MPI_Wtime();
}
}

// Разослать одно значение целого типа, то есть n, от нулевого процесса
// всем остальным процессам коммуникатора MPI_COMM_WORLD

MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
if (n == 0)
done = 1;
else
{ h = 1.0 / (double) n;
sum = 0.0;
for (i = myid + 1; i <= n; i += numprocs)
{ x = h * ((double)i - 0.5);
sum += f(x);
}
mypi = h * sum;

// Выполнить редукцию по сумме типа Double в программе 0-го процесса,
// с накоплением в pi значений по одному значению mypi от каждого из
// остальных процессов коммуникатора MPI_COMM_WORLD

MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);
if (myid == 0)
{
endwtime = MPI_Wtime();
printf("pi is approximately %.16f, Error of pi is %.16f\n",
pi, fabs(pi - PI25DT));
printf("wall clock time = %f\n", endwtime-startwtime);
fflush(stdout);
}
}
}
MPI_Finalize();
return;
}

```

ВОПРОСЫ К ЗАЧЕТУ

1. Что означает аббревиатура MPI в переводе на русский язык? Как следует понимать этот перевод?
2. Что такое MPI: стандарт, библиотека или что-то иное?
3. С какими языками программирования совместима библиотека MPI?
4. Какие 6 функций MPI относятся к базовому набору, достаточному для разработки любых приложений? Каковы их названия и назначение?
5. Чем отличается MPI от MPI SH? Является ли она свободно распространяемым программным средством? В каких операционных системах может работать?
6. Какие наиболее часто используемые возможности программы MPICH Configuration tool вы знаете?
7. Как узнать, на каких узлах кластера установлена базовая система запуска?
8. Как задать число процессов, запускаемых на выполнение в кластере?
9. Что необходимо сделать с выполняемым файлом параллельной программы, полученным после компиляции и компоновки в среде Microsoft Visual C++ или Microsoft Visual Studio, чтобы он мог быть запущен на выполнение во всех узлах кластера?
10. В каком порядке запускаются процессы на узлах кластера? Как изменить этот порядок?
11. Есть ли какая-либо зависимость между номерами процессов и цветом вывода результата в окне Output? Сколько всего цветов используется? Можно ли назначить вывод одним цветом? Если да, то как это сделать?
12. Как зафиксировать имя пользователя MPI и пароль в текущем сеансе работы с MPI? В текущем сеансе работы с операционной системой? Для всех последующих сеансов работы с операционной системой?
13. Как оставить результаты нескольких выводов в окне Output?
14. Как направить результаты, выводимые в окно Output, в файл? Будут ли они при этом выводиться в окно вывода?
15. Можно ли запускать на разных хостах файлы с разными именами? Если да, то как это сделать?
16. Будет ли достигнуто уменьшение времени выполнения параллельной программы, если на одном компьютере кластера запустить на выполнение более одного процесса? Если да, то в каком случае это возможно?
17. Назовите группы функций, на которые делятся все функции MPI в зависимости от числа процессов и объясните, чем различаются эти группы.
18. Что необходимо для начала работы с функциями библиотеки MPI и для завершения работы с ней?
19. Объясните аббревиатуру MPI_COMM_WORLD. Что обозначает этот идентификатор?
20. Зачем используется функция fflush(stdout)? Что может произойти, если ее не использовать?
21. Для чего нужны параметры argc и *argv[] в функции main? Объясните подробнее их назначение.
22. Почти все функции MPI имеют тип int. Какие значения они могут вернуть и какой смысл имеют эти значения?
23. Как узнать текущую версию MPI?
24. Что обозначает идентификатор класса ошибки MPI_ERR_TRUNCATE?
25. Чем отличается действие функций MPI_Finalize() и MPI_Abort()?
26. Как узнать количество процессов, запущенных на выполнение, и номер текущего процесса?
27. Можно ли узнать имя процессора, на котором выполняется текущая ветвь параллельной программы? Что означает константа MPI_MAX_PROCESSOR_NAME?
28. Как узнать текущее время средствами MPI? От какого момента в прошлом оно отсчитывается? Какой тип имеет? Как узнать единицу измерения времени?
29. Чем отличаются двухточечные взаимодействия от коллективных?
30. Чем отличается блокирующий режим выполнения функций от неблокирующего?
31. Учитывается ли в стандартном режиме обмена, получено ли сообщение приемником или нет? При стандартном обмене с блокировкой? Без блокировки?
32. Когда считается завершенной синхронная передача?
33. Когда считается завершенной буферизованная передача?
34. Как следует понимать согласованный режим обмена (по готовности)?
35. Какие режимы передачи согласуются с какими режимами приема?
36. Что такое тег сообщения? Каково его максимальное значение?
37. Приведите примеры типов данных, используемых в MPI.
38. Изобразите схему двухточечного обмена между процессами с указанием характерных особенностей.
39. Что такое джокеры? Как понимать аббревиатуры MPI_ANY_SOURCE, MPI_ANY_TAG?
40. Объясните, что такое статусная переменная. Какие сведения из нее можно получить?
41. Что такое дедлок? Нарисуйте схему с возможностью дедлока. Объясните, почему дедлок может не состояться.
42. Нарисуйте схему, при которой дедлок происходит всегда.
43. Что такое обмен по схеме двустороннего randеву?

44. Как узнать фактическую длину сообщения? Какие две функции для этого существуют и в чем разница между ними?
45. Можно ли узнать параметры сообщения до его фактического получения? Если да, то как это сделать?
46. Что такое хэнд-шейк сигнал?
47. Используется при буферизованном обмене системный буфер? Если да, то каким образом?
48. Объясните особенности выполнения коллективных взаимодействий.
49. Каким образом выполняется синхронизация различных ветвей параллельной программы?
50. Что такое барьеры? Поясните их действие графически.
51. Нарисуйте схему широковещательной рассылки данных функцией MPI_Bcast.
52. Как выполняется распределение блоков данных одинаковой длины по всем процессам группы с помощью соответствующей функции? Поясните графически.
53. Как выполняется сборка блоков данных одинаковой длины от разных процессов в корневом процессе? Поясните графически.
54. Объясните, в том числе с использованием графической иллюстрации, как выполняется сборка блоков данных разной длины от разных процессов в корневом процессе.
55. Какие вы знаете коллективные вычислительные операции над данными всех процессов коммутатора?
56. Какие предопределенные операции для функции редукции вы знаете?
57. Нарисуйте схему редукции с операцией MPI_SUM.
58. Какова максимальная производительность современных суперкомпьютеров?
59. Приведите примеры отраслей, в которых существует необходимость использования высокопроизводительных вычислительных систем.
60. Что такое 1 петафлопс, эксафлопс?
61. В чем заключается каскадная схема суммирования?
62. Объясните суть ленточного алгоритма умножения матриц.
63. В чем суть параллельного решения краевой задачи методом простой итерации с использованием явного итерационного алгоритма?
64. Закон Амдала. Графическая иллюстрация. Ускорение. Удельный вес скалярных операций. Параллелизм задачи.
65. Сетевой закон Амдала. Коэффициент сетевой деградации. Сетевое ускорение. Коэффициент утилизации.
66. Технические характеристики современных многопроцессорных вычислительных систем. Понятие о списке Top 500. Понятие об R_{max} , R_{peak} , Gflops.
67. Методы распараллеливания при вычислении суммы ряда. Учет производительности узла.

ЛИТЕРАТУРА

1. Ефимов С.С. Параллельное программирование: Учебное пособие. – Омск: «УниПак», 2009. – 400 с.
2. Ефимов С.С. Методические указания к лабораторным работам по курсу «Параллельные вычисления». – Омск: ОмГТУ, 2006. – 32 с.
3. Ефимов С. С. Распределенное программирование с использованием интерфейса передачи сообщений : учебник / С. С. Ефимов - Омск : ОмГТУ, 2007. - 384 с.
4. Воеводин В.В. Параллельные вычисления: Учеб. Пособие для вузов. – СПб.: БХВ-Петербург, 2002.