

Знакомство с технологией OpenMP

© М. Э. Абрамян, 2021

Обработка вложенных циклов

При выполнении заданий необходимо не только вывести требуемые результаты, но и отобразить дополнительную информацию в разделе отладки окна задачника, используя функции Show и ShowLine: время работы непараллельного варианта алгоритма (Non-parallel time), количество процессоров (num_procs), количество нитей (потоков) в параллельном варианте алгоритма (num_threads), количество слагаемых внутренних сумм, вычисленных в каждом потоке, и время, использованное на вычисления (указывается номер потока thread_num, количество слагаемых Count и время Thread time), общее время работы параллельного варианта алгоритма (Total parallel time), ускорение параллельного варианта по сравнению с непараллельным (Rate, вычисляется как отношение времени работы непараллельного варианта к общему времени работы параллельного варианта). Время выводится в миллисекундах.

Пример содержимого раздела отладки:

```
1> Non-parallel time: 350.87
2> num_procs: 2
3> num_threads: 2
4> thread_num: 1 Count: 2471976 Thread time: 165.48
5> thread_num: 0 Count: 2471964 Thread time: 167.14
6> Total parallel time: 168.53
7> Rate: 2.08
```

При выводе данных в параллельных разделах программы следует использовать директивы master и critical. Значение Thread time не должно включать время, использованное в каждом потоке для вывода информации. **Более предпочтительным вариантом** является сохранение нужных данных во вспомогательных массивах (размер которых равен числу потоков) и вывод этих данных *после выхода из параллельной области*.

Programming Taskbook - Электронный задачник по программированию [C++]

ОБРАБОТКА ВЛОЖЕННЫХ ЦИКЛОВ

Задание: OMPBegin4*

Выполняет: 3-Абрамян М. Э.

Результаты (F2) Цвет (F3) Режим (F4)

Данное задание включено в вариант 3693. Баллы: 2.

Ознакомительный запуск:
не выполнена ни одна из операций ввода-вывода.

Выход (Esc)

Ввод: 0 из 4 Вывод: 0 из 2 Тесты: 1 из 3

Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N \frac{1}{(\sum_{J=1}^N (J + (X + J)^{1/3}) / (2IJ - 1))}$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по полосам, вычисляя все слагаемые с четными значениями I в одном из потоков, а все слагаемые с нечетными значениями I — в другом.

Исходные данные

Данные для непараллельного варианта:	X = 0.7911	N = 1816
Данные для параллельного варианта:	X = 0.7413	N = 1820

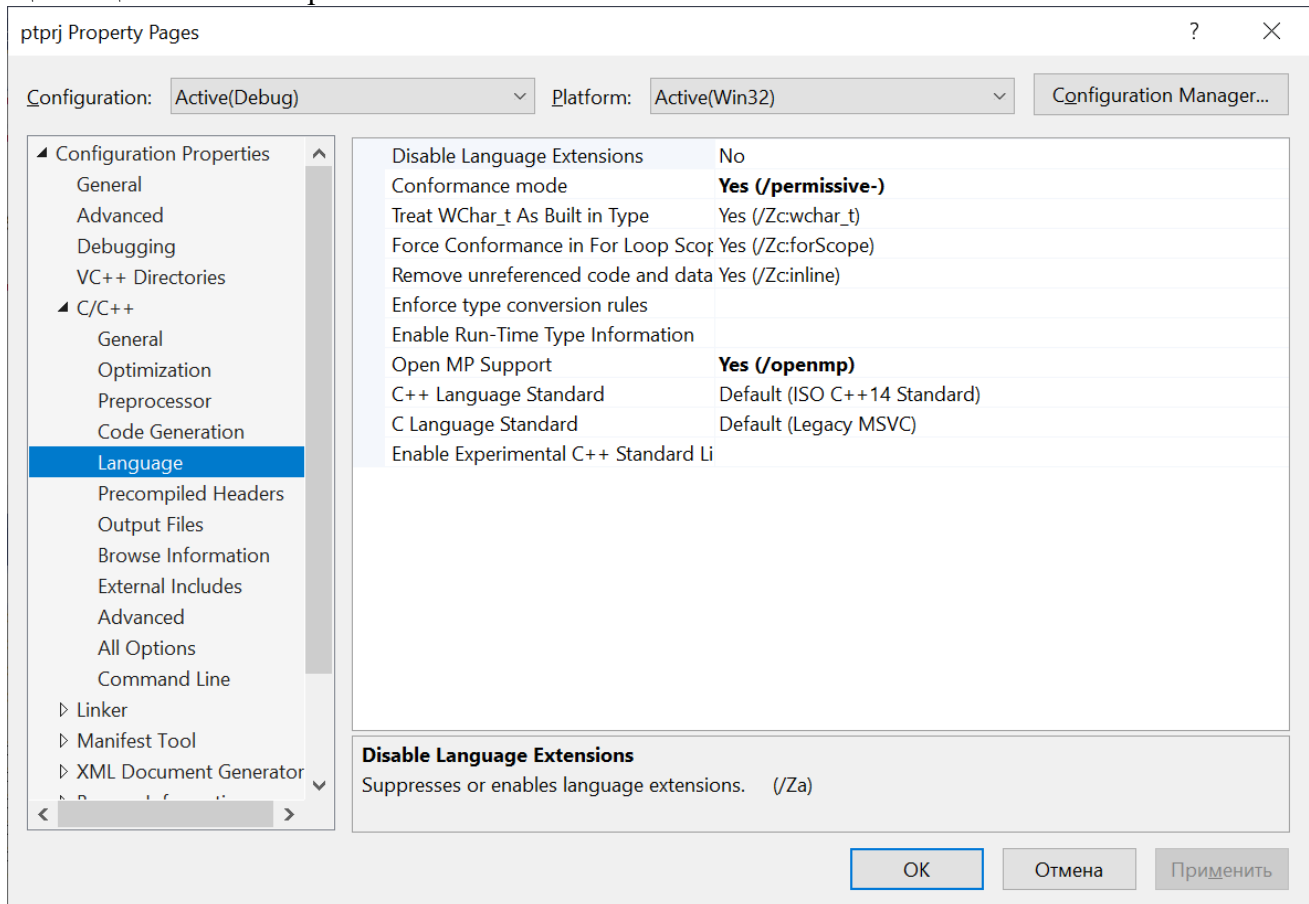
Пример результатов, выведенных в разделе отладки:

```
1> Non-parallel time: 288.85
2> num_procs: 2
3> num_threads: 2
4> thread_num: 1 Count: 2348556 Thread time: 134.81
5> thread_num: 0 Count: 2350089 Thread time: 132.73
6> Total parallel time: 138.80
7> Rate: 2.08
```

Пример верного решения

Результат для непараллельного варианта:	$F(X, N) =$	1395.2273
Результат для параллельного варианта:	$F(X, N) =$	1398.3116

Созданные для данных заданий заготовки программ уже содержат настройки, обеспечивающие поддержку OpenMP. В частности, для среды Visual Studio установлена соответствующая опция компилятора:



Текст заготовки имеет вид:

```
#include "pt4.h"
#include "omp.h"

void Solve()
{
    Task("OMPBegin4");
}
```

OMPBegin1. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=1}^I (J + (X + J)^{1/3})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *полосам*, вычисляя все слагаемые с четными значениями I в одном из потоков, а все слагаемые с нечетными значениями I — в другом.

OMPBegin2. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=1}^{I+N} (J + (X + J)^{1/3})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *полосам*, вычисляя все слагаемые с четными значениями I в одном из потоков, а все слагаемые с нечетными значениями I — в другом.

OMPBegin3. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^N (J + (X + J)^{1/3})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *полосам*, вычисляя все слагаемые с четными значениями I в одном из потоков, а все слагаемые с нечетными значениями I — в другом.

OMPBegin4. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{2N} (J + (X + J)^{1/3})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *полосам*, вычисляя все слагаемые с четными значениями I в одном из потоков, а все слагаемые с нечетными значениями I — в другом.

OMPBegin5. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=1}^I (J + \sin(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *обратным полосам*, разбивая набор значений I на полосы шириной 8 и распределяя их обработку следующим образом: в потоке 0 обрабатывается первый и последний элемент каждой полосы, в потоке 1 — второй и предпоследний элемент, и т. д. (в последнем потоке обрабатываются два средних элемента каждой полосы). Считать, что для параллельного варианта вычисления функции число N кратно 8.

OMPBegin6. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=1}^{I+N} (J + \sin(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *обратным полосам*, разбивая набор значений I на полосы шириной 8 и распределяя их обработку следующим образом: в потоке 0 обрабатывается первый и последний элемент каждой полосы, в потоке 1 — второй и предпоследний элемент, и т. д. (в последнем потоке обрабатываются два средних элемента каждой полосы). Считать, что для параллельного варианта вычисления функции число N кратно 8.

OMPBegin7. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^N (J + \sin(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *обратным полосам*, разбивая набор значений I на полосы шириной 8 и распределяя их обработку следующим образом: в потоке 0 обрабатывается первый и последний элемент каждой полосы, в потоке 1 — второй и предпоследний элемент, и т. д. (в последнем потоке обрабатываются два средних элемента каждой полосы). Считать, что для параллельного варианта вычисления функции число N кратно 8.

OMPBegin8. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный низкоуровневый, использующий функцию `omp_get_thread_num()` (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{2N} (J + \sin(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого использовать алгоритм распределения слагаемых по *обратным полосам*, разбивая набор значений I на полосы шириной 8 и распределяя их обработку следующим образом: в потоке 0 обрабатывается первый и последний элемент каждой полосы, в потоке 1 — второй и предпоследний элемент, и т. д. (в последнем потоке обрабатываются два средних элемента каждой полосы). Считать, что для параллельного варианта вычисления функции число N кратно 8.

OMPBegin9. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив `sections` и `section` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^I (J + \ln(1 + X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на две части (с диапазонами значений $1..K$ и $K+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. При указании диапазонов значений параметра I использовать формулу, выражающую зависимость K от N . Для вывода формулы воспользоваться соотношениями между площадями соответствующих геометрических фигур (прямоугольных треугольников или прямоугольных трапеций), получив в результате квадратное уравнение относительно неизвестной K .

OMPBegin10. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив `sections` и `section` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{I+N} (J + \ln(1 + X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на две части (с диапазонами значений $1..K$ и $K+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. При указании диапазонов значений параметра I использовать формулу, выражающую зависимость K от N . Для вывода формулы воспользоваться соотношениями между площадями соответствующих геометрических фигур (прямоугольных треугольников или прямоугольных трапеций), получив в результате квадратное уравнение относительно неизвестной K .

OMPBegin11. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив sections и section (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^N (J + \ln(1 + X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на две части (с диапазонами значений $1..K$ и $K+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. При указании диапазонов значений параметра I использовать формулу, выражающую зависимость K от N . Для вывода формулы воспользоваться соотношениями между площадями соответствующих геометрических фигур (прямоугольных треугольников или прямоугольных трапеций), получив в результате квадратное уравнение относительно неизвестной K .

OMPBegin12. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив sections и section (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{2N} (J + \ln(1 + X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на две части (с диапазонами значений $1..K$ и $K+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. При указании диапазонов значений параметра I использовать формулу, выражающую зависимость K от N . Для вывода формулы воспользоваться соотношениями между площадями соответствующих геометрических фигур (прямоугольных треугольников или прямоугольных трапеций), получив в результате квадратное уравнение относительно неизвестной K .

OMPBegin13. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив sections и section (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^I (J + (X + J)^{1/4})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на четыре части (с диапазонами значений $1..K_1$, $K_1+1..K_2$, $K_2+1..K_3$ и $K_3+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. Значения K_1 , K_2 , K_3 определить с помощью следующего вспомогательного алгоритма: вначале находится величина $K_0 = K / 4$, где K равно общему количеству слагаемых внутренних сумм; затем организуется перебор значений параметра I , в ходе которого в счетчике C накапливается общее количество слагаемых внутренних сумм, соответствующих перебираемым значениям I , пока значение C не превысит K_0 , после чего текущее значение параметра I записывается в очередную из переменных K_1 , K_2 , K_3 , а счетчик C сбрасывается в 0.

OMPBegin14. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив sections и section (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{I+N} (J + (X + J)^{1/4})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на четыре части (с диапазонами значений $1..K_1$, $K_1+1..K_2$, $K_2+1..K_3$ и $K_3+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. Значения K_1 , K_2 , K_3 определить с помощью следующего вспомогательного алгоритма: вначале находится величина $K_0 = K / 4$, где K равно общему количеству слагаемых внут-

ренных сумм; затем организуется перебор значений параметра I , в ходе которого в счетчике C накапливается общее количество слагаемых внутренних сумм, соответствующих перебираемым значениям I , пока значение C не превысит K_0 , после чего текущее значение параметра I записывается в очередную из переменных K_1, K_2, K_3 , а счетчик C сбрасывается в 0.

OMPBegin15. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив `sections` и `section` (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^N (J + (X + J)^{1/4})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на четыре части (с диапазонами значений $1..K_1, K_1+1..K_2, K_2+1..K_3$ и $K_3+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. Значения K_1, K_2, K_3 определить с помощью следующего вспомогательного алгоритма: вначале находится величина $K_0 = K / 4$, где K равно общему количеству слагаемых внутренних сумм; затем организуется перебор значений параметра I , в ходе которого в счетчике C накапливается общее количество слагаемых внутренних сумм, соответствующих перебираемым значениям I , пока значение C не превысит K_0 , после чего текущее значение параметра I записывается в очередную из переменных K_1, K_2, K_3 , а счетчик C сбрасывается в 0.

OMPBegin16. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании директив `sections` и `section` (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^{2N} 1/(\sum_{J=I}^{2N} (J + (X + J)^{1/4})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков. Для этого разбить внешнюю сумму на четыре части (с диапазонами значений $1..K_1, K_1+1..K_2, K_2+1..K_3$ и $K_3+1..N$ параметра I), содержащие примерно одинаковое количество слагаемых внутренних сумм. Значения K_1, K_2, K_3 определить с помощью следующего вспомогательного алгоритма: вначале находится величина $K_0 = K / 4$, где K равно общему количеству слагаемых внутренних сумм; затем организуется перебор значений параметра I , в ходе которого в счетчике C накапливается общее количество слагаемых внутренних сумм, соответствующих перебираемым значениям I , пока значение C не превысит K_0 , после чего текущее значение параметра I записывается в очередную из переменных K_1, K_2, K_3 , а счетчик C сбрасывается в 0.

OMPBegin17. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании статического варианта директивы `for` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=1}^I (J + \cos(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin18. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании статического варианта директивы `for` (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=1}^{I+N} (J + \cos(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin19. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании статического варианта директивы for (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^N (J + \cos(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin20. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании статического варианта директивы for (число потоков равно 2). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{2N} (J + \cos(X + J))/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin21. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании динамического варианта директивы for (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^I (J + (X + J)^{1/5})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin22. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании динамического варианта директивы for (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{I+N} (J + (X + J)^{1/5})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin23. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании динамического варианта директивы for (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^N (J + (X + J)^{1/5})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.

OMPBegin24. Реализовать два варианта вычисления функции $F(X, N)$: непараллельный и параллельный, основанный на использовании динамического варианта директивы for (число потоков равно 4). Формула для функции F :

$$F(X, N) = \sum_{I=1}^N 1/(\sum_{J=I}^{2N} (J + (X + J)^{1/5})/(2IJ - 1))$$

В разделе отладки вывести дополнительную информацию, описание которой приведено в преамбуле к данной группе заданий. Обеспечить возможно большее ускорение параллельного варианта за счет равномерной загрузки потоков.