

Лабораторная работа №2

Распараллеливание алгоритмов с использованием OpenMP

Цель работы: изучить способы создания параллельных программ для симметричных мультипроцессоров с использованием стандарта OpenMP.

Теоретические материалы

Директива распараллеливания циклов.

Для того, чтобы распределить выполнение цикла по имеющимся ядрам, используется следующая директива:

```
#pragma omp for [clause ...] newline for loop
```

Распределение итераций в директиве for регулируется параметром (clause) schedule:

- static – итерации делятся на блоки по chunk итераций и статически разделяются между потоками; если параметр chunk не определен, итерации делятся между потоками равномерно и непрерывно;
- dynamic – распределение итерационных блоков осуществляется динамически (по умолчанию chunk=1);
- guided – размер итерационного блока уменьшается по экспоненциальному закону при каждом распределении;
- chunk определяет минимальный размер блока (по умолчанию chunk=1);
- runtime – правило распределения определяется переменной OMP_SCHEDULE (при использовании runtime параметр chunk задаваться не должен).

Директива параллельных секций

Для выделения отдельных фрагментов кода в параллельные области используется директива **sections**. Отметим ряд свойств этой директивы:

- каждый фрагмент выполняется однократно (директива **section**);
- разные фрагменты выполняются разными потоками;
- завершение директивы по умолчанию синхронизируется;
- директивы **sections** должны использоваться только в статическом контексте.

Директива непараллельных секций

Чтобы выполнить блок только один раз, используется директива single

```
#pragma omp single [name]
```

<структурный блок >

Для запрета конкурентного выполнения блока кода используется директива critical:

#pragma omp critical [name]

<структурный блок >

Отличия этих директив друг от друга состоит в том, что блок single выполняется только единожды каким-либо потоком, а блок critical – всеми потоками, но только одним одновременно.

Задание на лабораторную работу

Распараллельте заданный алгоритм с использованием OpenMP. Оцените ускорение многопоточного выполнения. При создании параллельной программы требуется использовать имеющиеся инструменты OpenMP: параллельный цикл и параллельную секцию.

1. Распараллельте алгоритм с использованием параллельного цикла OpenMP. Оцените время выполнения
2. Распараллельте этот же алгоритм с использованием механизма параллельных секций OpenMP. Тоже оцените время выполнения.
3. Оцените время выполнения последовательного варианта.

Сравните быстродействие во всех трёх случаях и поясните результаты.

Варианты заданий

Вариант	Операции с векторами
1	$C = A + B$
2	$C = A - B * x$
3	$a = (B * C)$
4	$a = \text{MAX}(B) * C$
5	$b = \text{MIN}(A + C)$
6	$A = B * \text{MIN}(C)$
7	$A = B + C - D * e$
8	$C = A - B + D$
9	$d = \text{MAX}(A + B + C)$

Вариант	Операции с векторами
10	$d = ((A + B) * C)$
11	$d = (A * (B + C))$
12	$d = (A * B) + (C * B)$
13	$d = (B * C) - (A * B) + (C * B)$
14	$E = A + B + C - D * e$
15	$E = A + C - B * e + D$
16	$e = ((A + B) * (C + D))$

Пояснения: A, B, C, D – векторы некоторого размера (одинакового для всех в рамках одного варианта); e, x – некоторые скалярные величины, задаваемые пользователем.