**Лабораторная работа №5**

**Распределение вычислительной нагрузки**

**Цель:** изучить основные особенности распределения вычислительной нагрузки в OpenMP на примере использования в рамках языка С++.

Для получения **теоретических сведений** настоятельно рекомендуется при домашней подготовке изучить материалы, представленные в списке литературы в конце разработки, а также прочие материалы по тематике лабораторной работы, представленные в открытых источниках.

Далее следует краткий конспект материала, приведенного в данных источниках, в конце включающий короткие примеры фрагментов программ.

**1. Распределение вычислительной нагрузки между потоками (распараллеливание по задачам при помощи директивы sections)**

Широко встречающейся является ситуация, когда для решения задачи необходимо выполнить разные процедуры, которые или полностью не зависят друг от друга, или являются слабо связанными. Такие процедуры можно выполнить параллельно; такой подход именуется распараллеливанием по задачам. Для этого в OpenMP для параллельного фрагмента программы, можно выделять параллельно выполняемые программные секции (директива sections).

Формат директивы sections имеет вид:

#pragma omp sections [<параметр> ...]

{

#pragma omp section

<блок\_программы>

#pragma omp section

<блок\_программы>

}

При помощи директивы sections выделяется программный код, который далее будет разделен на параллельно выполняемые секции. Директивы section определяют секции, которые могут быть выполнены параллельно (для первой по порядку секции директива section не является обязательной).

В зависимости от взаимного сочетания количества потоков и количества определяемых секций, каждый поток может выполнить одну или несколько секций (вместе с тем, при малом количестве секций некоторые потоки могут оказаться без секций и окажутся незагруженными). Использование секций достаточно сложно поддается масштабированию (настройке на число имеющихся потоков). Кроме того, в соответствии со стандартом, порядок выполнения программных секций не определен.

В качестве параметров директивы sections могут использоваться:

- private (list)

- firstprivate (list)

- lastprivate (list)

- reduction (operator: list)

- nowait

**Лабораторные задания**

**Задание.** Модифицировать программы, составленные в Л.Р. №№3,4, распределив вычислительную нагрузку по секциям (как минимум двумя способами). Измерять время работы программы для тех же значений параметров, что были использованы при выполнении Л.Р. №3. **Результаты сравнить и занести в отчёт.**

**Требования к сдаче работы**

1. При домашней подготовке изучить теоретический материал по тематике лабораторной работы, представленный в списке литературы ниже, выполнить представленные примеры, занести в отчёт результаты выполнения.
2. Продемонстрировать выполнение лабораторных заданий.
3. Ответить на контрольные вопросы.
4. Показать преподавателю отчет.

**Литература**

1. Спецификации стандарта OpenMP (на английском языке):

<http://openmp.org/wp/openmp-specifications/>

2. Материалы, представленные на сайте intuit.ru в рамках курса «Intel Parallel Programming Professional (Introduction)»:

<http://old.intuit.ru/department/supercomputing/ppinteltt/4/>

3. С.А. Лупин, М.А. Посыпкин Технологии параллельного программирования. – М.: ИД «ФОРУМ»: ИНФРА-М, 2011. – С. 119-145. *(Глава, посвященная OpenMP)*

4. Канг Су Гэтлин, Пит Айсенси - OpenMP и C++:

<http://msdn.microsoft.com/ru-ru/library/dd335940.aspx>

5. API OpenMP C и C++

<http://msdn.microsoft.com/ru-ru/library/8y6825x5.aspx>