Национальный исследовательский ядерный университет «МИФИ» (Московский Инженерно–Физический Институт)
Кафедра №42 «Криптология и кибербезопасность»

Лабораторная работа №1:
«Введение в параллельные вычисления. Технология OpenMP»

Предельная дата защиты: 05.10.2023

Цель работы

Приобрести базовые навыки теоретического и экспериментального анализа высокопроизводительных параллельных алгоритмов, построения параллельных программ.

Ход работы

1. Настроить рабочую среду для построения параллельных программ, реализованных при помощи стандарта OpenMP. Определить версию OpenMP, поддерживаемую используемым компилятором;
2. Изучить приложенный алгоритм поиска максимального элемента в массиве. Оценить его временну́ю сложность. Оценить ускорение и эффективность параллельных вычислений. Построить блок-схему алгоритма;
3. Построить параллельную программу, убедиться в её работоспособности;
4. Экспериментально определить временну́ю сложность (последовательного) алгоритма двумя способами: с помощью прямого подсчёта выполненных операций сравнения и путём измерения времени работы программы.
5. Экспериментально определить ускорение и эффективность параллельных вычислений путём измерения времени работы программы при использовании 1, 2,. . . процессоров;
6. Сравнить экспериментальные показатели ускорения и эффективности с теоретическими на графике;
7. Оформить отчёт.

Рекомендации и указания

1. Во многих инструментах поддержка OpenMP по умолчанию отключена. Её необходимо включить явно: в настройках проекта или при помощи соответствующего флага в командной строке (обычно "/openmp" или "-fopenmp"). Важно: параллельные программы на OpenMP, использующие только директивы предпроцессора, могут быть построены компилятором без поддержки OpenMP. В этом случае получится «обычная», последовательная программа. Поэтому при построении проекта важно проверить, что дополнительные потоки действительно создаются. Для этого можно вывести на консоль текстовое сообщение или номер текущего потока в одной из параллельных областей.
2. Заголовочный файл "omp.h" нужен, если используются функции OpenMP. Если программа использует только директивы предпроцессора, необходимости в его подключении нет.
3. При экспериментальном анализе (определении времени работы алгоритма и числа операций сравнения) необходимо запустить алгоритм не менее 10 раз и усреднить полученные результаты. Важно проследить, чтобы генератор случайных чисел при этом инициализировался разными значениями. Важно, что подсчёт операций сравнения влияет на время работы алгоритма, поэтому эти два эксперимента необходимо проводить отдельно.
4. При экспериментальном анализе ускорения и эффективности необходимо запустить алгоритм не менее 10 раз для каждого числа процессоров. Используемые при этом 10 массивов должны быть различными. Но: для каждого числа процессоров эти массивы должны быть одинаковыми. То есть: 10 различных массивов на одном процессоре, 10 таких же массивов на двух, 10 таких же массивов на трёх, и т. д.

Оформление отчёта

1. Титульный лист: название института, название лабораторной работы, имя, фамилия, номер группы, год,...
2. Описание используемой рабочей среды: модель процессора, объём и тип оперативной памяти, версия и разрядность операционной системы, используемая среда разработки, поддерживаемая ею версия OpenMP;
3. Анализ приведённого алгоритма: описание принципа его работы. Блок-схема алгоритма. Для каждой директивы OpenMP указать: её смысл, область программного кода, на которую она распространяется, какую роль она играет в программе (и что бы было, если бы её не было);
4. Графики: время работы, ускорение и эффективность в зависимости от числа процессоров. На графиках сравнить теоретические оценки с экспериментальными;
5. Заключение: краткое описание проделанной работы;
6. Приложение: использованные в работе программные коды;
7. Приложение: таблицы с теоретическими результатами и результатами вычислительных экспериментов.