

Задания к лабораторной работе
«Параллельные вычисления в разделяемой памяти»
по предмету «Параллельное программирование»

Цель работы: знакомство с основными приёмами параллельного программирования и исследования производительности параллельных программ.

Порядок выполнения работы.

1. Реализация параллельной программы первым способом (указан в задании).
2. Добавление в программы кода для анализа производительности программы п.1.
Проведение вычислительных экспериментов с целью установления зависимости ускорения и параллельной программы от размера решаемой задачи. Оценка эффективности распараллеливания.
3. Реализация параллельной программы вторым способом (указан в задании).
4. Добавление в программы кода для анализа производительности программы п.3.
Проведение вычислительных экспериментов с целью установления зависимости ускорения параллельной программы от размера решаемой задачи. Оценка эффективности распараллеливания.
5. Оформление отчёта. Отчет должен включать: листинги программ, результаты экспериментов в табличной форме, обработанные результаты экспериментов в графической форме, выводы возможности и целесообразности распараллеливания задач исследуемыми методами, согласованность наблюдаемого поведения программ с теоретическими оценками производительности (качественными или количественными).

Варианты заданий.

1. Требуется реализовать параллельный алгоритм построчного вычисления произведения квадратных матриц в разделяемой памяти. Сравнить производительность OpenMP-реализации и реализации, использующей API операционных систем.
2. Требуется реализовать параллельный алгоритм постолбцового вычисления произведения квадратных матриц в разделяемой памяти. Сравнить производительность OpenMP-реализации и реализации, использующей API операционных систем.
3. Требуется реализовать параллельный алгоритм поэлементного вычисления произведения квадратных матриц в разделяемой памяти. Сравнить производительность OpenMP-реализации и реализации, использующей API операционных систем.
4. Требуется реализовать параллельный алгоритм построчного вычисления произведения квадратных матриц в архитектуре управляющий-рабочие (разделяемая память). Сравнить производительность реализации на основе шаблона Taskbag и реализации, использующей OpenMP.
5. Требуется реализовать параллельный алгоритм постолбцового вычисления произведения квадратных матриц в архитектуре управляющий-рабочие (разделяемая память). Сравнить производительность реализации на основе шаблона Taskbag и реализации, использующей OpenMP.

6. Требуется реализовать параллельный алгоритм поэлементного вычисления произведения квадратных матриц в архитектуре управляющий-рабочие (разделяемая память). Сравнить производительность реализации на основе шаблона Taskbag и реализации, использующей OpenMP.
7. Требуется реализовать рекурсивный параллельный алгоритм вычисления интеграла функции по методу адаптивной квадратуры. В данном методе интеграл функции на интервале $[a, b]$ принимается равным $S = (f(a) + f(b)) / 2(b - a)$, если $|S - S_1 - S_2| < E$; где E – точность интегрирования, $S_1 = (f(a) + f(m)) / 2(m - a)$, $S_2 = (f(m) + f(b)) / 2(b - m)$, $m = (a + b) / 2$. В противном случае он ищется как сумма интегралов на отрезках $[a, m]$ и $[m, b]$ рекурсивным обращением к описанной процедуре. Сравнить производительность OpenMP- реализации и реализации, использующей API операционных систем.
8. Требуется реализовать итеративный вариант параллельного алгоритма вычисления интеграла функции в котором область интегрирования делится на фиксированное число интервалов. Сравнить производительность реализации на основе шаблона Taskbag и реализации, использующей OpenMP.
9. Реализовать параллельный вариант алгоритма быстрой сортировки. Сравнить производительность реализации на основе шаблона Taskbag и реализации, использующей OpenMP.
10. Реализовать параллельный вариант алгоритма сортировки слиянием. Сравнить производительность реализации на основе шаблона Taskbag и реализации, использующей OpenMP.