

5 Лабораторная работа №2. «Исследование эффективности параллельных библиотек для С-программ»

5.1 Порядок выполнения работы

1. В исходном коде программы, полученной в результате выполнения лабораторной работы №1, нужно на этапах Map и Merge все циклы с вызовами математических функций заменить их векторными аналогами из библиотеки «AMD Framewave» (<http://framewave.sourceforge.net>). При выборе конкретной Framewave-функции необходимо убедиться, что она помечена как MT (Multi-Threaded), т.е. распараллеленная. Полный перечень доступных функций находится по ссылке: http://framewave.sourceforge.net/Manual/fw_section_060.html#fw_section_060. Например, Framewave-функция min в списке поддерживаемых технологий имеет только SSE2, но не MT.

Примечание: выбор библиотеки Framewave не является обязательным, можно использовать любую другую параллельную библиотеку, если в ней нужные функции распараллелены.

2. Добавить в начало программы вызов Framewave-функции SetNumThreads(M) для установки количества создаваемых параллельной библиотекой нитей, задействуемых при выполнении распараллеленных Framewave-функций. Нужно число M следует устанавливать из параметра командной строки (argv) для удобства автоматизации экспериментов.
3. Скомпилировать программу, не применяя опции автоматического распараллеливания, использованные в лабораторной работе №1. Провести эксперименты с полученной программой для тех же значений N_1 и N_2 , которые использовались в лабораторной работе №1, при $M = 1, 2, \dots, K$, где K – количество процессоров (ядер) на экспериментальном стенде.
4. Сравнить полученные результаты с результатами лабораторной работы №1: на графиках показать, как изменилось время выполнения программы, параллельное ускорение и параллельная эффективность.
5. Написать отчёт о проделанной работе.

6. Подготовиться к устным вопросам на защите.
7. **Необязательное задание №1** (для получения оценки «четыре»). Исследовать параллельное ускорение для различных значений $M > K$, т.е. оценить накладные расходы при создании чрезмерного большого количества нитей. Для иллюстрации того, что программа действительно распараллелилась, привести график загрузки процессора (ядер) во время выполнения программы при $N = N_2$ для всех использованных M . Для получения графика можно как написать скрипт, так и просто сделать скриншот диспетчера задач, указав на скриншоте моменты начала и окончания эксперимента (в отчёте нужно привести текст скрипта или название использованного диспетчера).
8. **Необязательное задание №2** (для получения оценки «пять»). Это задание выполняется только после выполнения предыдущего пункта. Используя закон Амдала, рассчитать коэффициент распараллеливания для всех экспериментов и привести его на графиках. Прокомментировать полученные результаты.

5.2 Состав отчета

1. Титульный лист с названием вуза, ФИО студентов и названием работы.
2. Содержание отчета (с указанием номера страниц и т.п.).
3. Краткая характеристика использованного для проведения экспериментов процессора, операционной системы и компилятора (официальное название, номер версии/модели, разрядность, количество ядер, ёмкость ОЗУ и т.п.).
4. Описание особенностей конфигурации использованной параллельной библиотеки, включая описание последовательности шагов, принятых для установки библиотеки, и использованных опций компиляции.
5. Полный текст полученной параллельной программы, а также текст всех скриптов, использованных для компилирования программы и проведения экспериментов.

6. Графики функций времени выполнения использованных программ, а также графики параллельного ускорения и параллельной эффективности для разных N и M (допускается совмещать несколько графиков в одной системе координат).
7. Подробные выводы с анализом приведённых графиков и полученных результатов. Отчёт предоставляется на бумажном носителе или на флешке.

5.3 Подготовка к защите

1. Уметь объяснить каждую строку программы, представленной в отчёте.
2. Знать о назначении всех использованных в работе ключей компиляции.
3. Знать материал лекции №2.
4. Взять с собой все нужные файлы для демонстрации работы программы.