**Задания на разработку блочных и параллельных программ перемножения блочных треугольных матриц**

**Описание основных элементов задания**

Блочная верхнее-треугольная матрица имеет вид:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| А1,1 | А1,2 | А1,3 | … | А1,N |
| 0 | А2,2 | А2,3 | … | А2,N |
| ... | … | … | … | … |
| … | … | … | … | … |
| 0 | 0 | 0 | … | АN,N |

При этом, диагональные блоки являются верхне-треугольными матрицами. Блочная нижнее-треугольная матрица выглядит аналогично.

Обычные треугольные матрицы принято хранить в памяти в виде одномерного массива, в котором подряд выписываются либо строки, либо столбцы. Блочная треугольная матрица также должна храниться в виде одномерного массива, в котором подряд выписываются блочные строки или блочные столбцы, кроме нулевых блоков. При этом, элементы каждого отдельно взятого блока должны размещаться, согласно стандарту языка Си, построчно.

Таким образом, при размещении указанной выше блочной верхнее-треугольной матрицы поблочно построчно блоки должны располагаться в памяти в следующей последовательности:

А1,1, А1,2 ,…, А1,N , А2,1 , …, А2,N ,…, АN,N

При размещении этой же матрицы поблочно по столбцам блоки должны располагаться в памяти в следующей последовательности:

А1,1 , А1,2 , А2,2 , А1,3 , …, А1,N ,…, АN,N

Симметричная матрица должна размещаться в памяти либо как верхне-треугольная, либо как нижне-треугольная.

Смысл задания – увидеть, как выжимается производительность программы за счет перехода к блочному коду (данный прием называется tiling), а также за счет блочных размещений матриц в оперативной памяти. И в том и в другом случае ускорение происходит за счет минимизации кэш-промахов. Стоит отметить, что разные компиляторы могут генерировать код разной производительности.

**Требования к программе и отчету**

**Матрицы необходимо перемножать поблочно:** блочные строки умножаются на блочные столбцы (псевдо скалярное умножение).

Размер матриц 2880х2880.

Перемножение матриц необходимо произвести для разных размеров блоков (одинаковых для обеих перемножаемых матриц).

Размеры блоков: 1, 6, 10, 15, 20, 24, 30, 36, 40, 60, 72, 80, 96, 120, 144, 160, 180, 240, 360, 480, 720.

В отчете должна присутствовать следующая информация:

* описание характеристик компьютера (процессора и памяти), влияющих на быстродействие.

Пример:

Processor: Intel® Core™ i5-6600 CPU @ 3.30GHz; 4 cores; no hyperthreading;

L3: 6 MB (shared);

L2: 256 kB (split);

L1: 32 kB instruction cache, 32 kB data cache (split);

RAM: DDR4, 32 GB, clock speed: 2133 MHz.

* описание характеристик операционной системы, языка программирования и компилятора.

Пример:

Programming language: C++; (обязательно использовать именно этот язык)

Compiler: GСС v. 6.3.0, compiler option: -Ofast;

GNU OpenMP v. 6.3.0;

OS: Debian 9 (Linux);

* таблицы с результатами численных экспериментов. В таблице присутствует два столбца, в первом из которых, указываются размеры блоков (по возрастанию), а во втором сколько времени работает программа для соответствующего блока.
* графики зависимости времени счета от размерности блоков. Ось x – размеры блоков, y – время, расстояние между точками по x – 20.

Проверить корректность (правильность) программы умножения матриц (результаты необходимо сверять с результатами **не блочного** умножения матриц).

Программа должна быть написана на языке программирования C++ и собрана двумя различными компиляторами, для двух вещественных типов данных. Запускать нужно три версии программы:

* последовательное выполнение - при перемножении блоков не используется параллельности;
* параллельное 1 – параллельно выполняется умножение различных пар блоков;
* параллельное 2 – параллельно выполняется умножение каждого отдельного блока;

**Инструкция по составлению отчета**

Отчет по форме должен удовлетворять требованиям ГОСТ 7.32 2001. При этом обязательно должны быть включены следующие структурные элементы:

* титульный лист;
* реферат;
* основная часть;
* заключение.

Решение о включении остальных структурных элементов отчета принимается автором.

Основная часть должна содержать:

* раздел с описанием архитектуры компьютера, на котором проводились эксперименты;
* раздел с описанием экспериментов для первого выбранного автором компилятора;
* раздел с описанием экспериментов для второго выбранного автором компилятора;
* раздел, содержащий сравнительный анализ двух компиляторов, на примере исполнения кода, реализующего решение поставленной задачи.