**Лабораторное задание №3**

**Описание задачи:**

Реализовать метод решения системы линейных алгебраических уравнений (СЛАУ) – метод Гаусса.

**Описание метода:**

Путем применения элементарных преобразований над расширенной матрицей, описывающей систему (матрица коэффициентов + столбец свободных членов):

1) перемена мест двух строк или двух столбцов в данной матрице;  
2) умножение строки (или столбца) на произвольное число, отличное от нуля;  
3) прибавление к одной строке (столбцу) другой строки (столбца), умноженной на некоторое число.

привести систему к эквивалентной системе верхнетреугольного вида (прямой ход), из которой затем последовательно, начиная с последней строки, найти все переменные (обратный ход).

Прямой ход:

Обратный ход:

Из последнего ненулевого уравнения выражаем базисную переменную через небазисные и подставляем в предыдущие уравнения. Повторяя эту процедуру для всех базисных переменных, получаем решение.

Программа ***sle\_solve.cpp*** последовательно решает данную задачу, сперва для матрицы размером 3x3 для проверки корректности вычислений, сравнивая полученное решение с рассчитанными вручную значениями, затем на матрице большего размера (устанавливается через task\_size).

**Задание:**

1. Необходимо собрать проект с исходным файлом ***sle\_solve.cpp*** и запустить собранный исполняемый файл. Оценить время работы программы и корректность ее работы. Зафиксируйте это в отчете.

2. С помощью инструментария Advisor необходимо получить метрики исполнения программы в последовательном режиме: GFLOPs, количество используемых потоков, тип векторных команд, определить хотспоты, построить roofline, сохранить snapshot.

3. Максимально ускорить выполнение программы, убедиться в корректности ее работы (анализ в Inspector), построить совместные roofline для разных решений, оценить во сколько произошло ускорение. Для параллельной версии программы так же проанализировать эффективность выполнения через VTune. Для успешной сдачи ЛР необходимо добиться хотя бы 2-х кратного ускорения выполнения программы.

4. Подготовьте отчет (разделы ниже) и продемонстрируйте работу приложений и их характеристики преподавателю.

**Подготовьте отчет со следующими разделами:**

1. Опишите каким образом Вы проводили оптимизацию последовательно исполняемого приложения (смена последовательности выполнения циклов, разбивка циклов на подциклы, использование SoA вместо AoS и наоборот, раскручивание циклов). Оцените влияние последовательности доступа к данным и их выравнивания в памяти на векторизацию.
2. Какие значения основных метрик производительности последовательно исполняемого приложения в Intel Advisor Вы получили (производительность в GFLOPs, пропускная способность памяти, использование векторных инструкций, roofline модель). Как изменились метрики при введении векторизации? Представьте сравнение метрик производительности последовательной работы с приложением, в котором используется «ручная» оптимизация.
3. Проведите прогнозирование ускорения выполнения программы за счет введения annotations в Intel Advisor. Осуществите введение параллелизма в последовательно исполняемую программу с использованием программной модели OpenMP, проведите анализ утилизации ресурсов процессора в Intel VTune, сравните нескольких roofline между собой (параллельное исполнение, последовательное исполнение, «ручная» оптимизация) и подтверждение выполнения прогноза ускорения в Intel Advisor. Проверьте корректность выполнения параллельной программы (Intel Inspector).