Лабораторное задание №4

Описание задачи:

Поиск кратчайших путей между всеми парами вершин в ориентированном графе. На вход алгоритма подается матрица смежности W. Эта матрица задает веса ребер между парами соединенных вершин. В случае, когда между вершинами i,j ребро отсутствует, $W_{i,j} = \infty$. Матрица W – квадратная, размера $n \times n$.

Одна итерация алгоритма заключается в определении минимального пути до остальных вершин от текущей, путем добавления одной дополнительной (промежуточной) вершины в этот путь. На очередной итерации алгоритма будем иметь матрицу L весов кратчайших путей, тогда обновленный путь из вершины i в j, L'_{ii} будет лежать через новую вершину k:

$$L'_{ij} = \min(L_{ij}, 0 < k < n(L_{ik} + W_{kj}))$$

Таким образом, для получения всех возможных путей длиной n-1 (проходит через все остальные вершины), необходимо выполнить n-2 итераций.

Задание:

- 1. Необходимо собрать проект с исходным файлом *graph.cpp* и запустить собранный исполняемый файл. Оценить время работы программы и корректность ее работы. Зафиксируйте это в отчете. Оцените сложность алгоритма.
- 2. С помощью инструментария Advisor необходимо получить метрики исполнения программы в последовательном режиме: GFLOPs, количество используемых потоков, тип векторных команд, построить roofline, зафиксировать список хотспотов, выделить ключевые метрики, сохранить snapshot.
- 3. Максимально ускорить выполнение программы, убедиться в корректности ее работы, построить совместные roofline для разных решений, оценить во сколько произошло ускорение. Для сдачи лабораторной работы на полный балл необходимо получить ускорение для последовательной версии программы (без добавления параллелизма, за счет оптимизаций), а затем и с добавлением параллелизма. Зафиксировать оба шага в отчете.
- 4. Оценить эффективность параллельного алгоритма: через отчет в VTune, так же оценить масштабируемость задачи. Проверить наличие проблем в параллельной реализации через Intel Inspector.
- 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 (ОМР против ручного распараллеливания), последовательное исполнение, «ручная» оптимизация) и подтверждение выполнения прогноза ускорения в Intel Advisor. Проверьте корректность выполнения параллельной программы.