**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра информационных систем управления**

Веренич Владислав Николаевич

Этапы разработки параллельных алгоритмов Разработка многопоточных приложений на языке С++

Отчет по лабораторной работе № 3

«Распределенные и параллельные системы»

студента 3 курса 12 группы

**Преподаватель**

**Кондратьева О.М.**

**Минск 2022**

Этапы разработки параллельных алгоритмов:

**Этап 1.** *Разделение вычислений на независимые части.*

В качестве базовой операции можно выбрать умножение одной из строк матрицы на вектор. Такой выбор удовлетворяет двум основным критериям декомпозиции вычислений: независимость и равная трудоемкость.

**Этап 2.** *Выделение информационных зависимостей.*

Подзадачи независимо друг от друга вычисляют один из элементов вектора; после завершения всех подзадач необходимо лишь записать все полученные элементы в новый вектор.

**Этап 3.** *Масштабирование набора подзадач.*

Здесь требуется определить необходимую (или доступную) для решения задачи вычислительную систему и выполнить распределение имеющегося набора подзадач между процессорами системы. В нашем случае необходимо распределить n подзадач между p вычислительными элементами, когда количество подзадач значительно превышает количество вычислительных элементов. Для сокращения количества подзадач необходимо выполнить укрупнение (агрегацию) вычислений. Будем объединять отдельные строки матрицы в последовательные группы строк, то есть использовать ленточную схему на непрерывной основе. Получим p подзадач, каждая из которых находит n/p элементов результирующего вектора.

Замечу, что этот этап не меняется при разработке параллельной реализации модели делегирования 1 и модели делегирования 2.

**Этап 4.** *Распределение подзадач между вычислительными элементами.*

Распределение нагрузки обычно выполняется операционной системой автоматически. Кроме того, данный этап распределения подзадач между процессорами является избыточным, если количество подзадач совпадает с числом имеющихся вычислительных элементов.