**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

Лабораторная работа №4

**по дисциплине**

**«Распределенные программные системы и технологии»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3308 |  | Сысоев В.Б. |
| Преподаватель |  | Пазников А.А. |

Санкт-Петербург

2018

# 1. Задание

1. Разработать параллельную MPI-программу моделирования стационарного распределения тепла в тонкой стальной пластине.
2. Провести эксперименты и построить графики зависимости ускорения и коэффициента эффективности от количества потоков, оценить масштабируемость, построить оценки вычислительной и коммуникационной сложности.
3. Сформулировать рекомендации по выбору способа декомпозиции элементов, размера расчётной сетки и других параметров параллельной программы.

**Замечания:**

* Задача стационарного распределения тепла описывается уравнением Пуассона с заданными граничными условиями.
* Пластина с трёх сторон окружена водяным паром (температура 100°С), а с четвёртой стороны граничит со льдом (температура 0°С). Сверху и снизу пластины находится теплоизоляционный слой.
* При реализации параллельной программы необходимо использовать оптимальный способ декомпозиции расчётной области.
* Размеры пластины: a = 1 м, b = 2 м.

# 2. Описание решения задачи

Реализованный в лабораторной работе алгоритм является параллельным итерационным методом Зейделя с «красно-черным» упорядочиванием узлов, описанным в [1].

При заданном уравнении:

Приближенно решаемом на сетке с шагом h, где приближенное решение в узле сетки с индексами i. j на k-м шаге обозначено как . Значения в узлах красного цвета вычисляются по формуле:

Значения узлов черного цвета вычисляются как:

Учитывая условие задачи, слагаемое в данных формулах будет равняться 0.

Критерием остановки итераций служит сходимость максимального модуля невязки с заданной точностью, где невязка для каждого узла сетки вычисляется следующим образом:

здесь с помощью индексов обозначаются значения функций в соответствующих узлах сетки. Стоит отметить, что в данной задаче значение fi,j будет всегда равно 0.

Вычислительная сложность каждой итерации алгоритма будет пропорциональна квадрату количества узлов одной стороны сетки. При параллельной реализации, коммуникационная сложность будет равна сумме периметров кусков сетки обрабатываемых каждым процессом, отсюда можно сделать вывод о том, что оптимальной декомпозицией будет деление сетки на куски, максимально приближенные по пропорциям к квадрату.

Учитывая данные заключения, параллельный алгоритм был реализован с использованием двумерной виртуальной топологии и дифференцированного обмена сообщениями. Каждый шаг итерации состоит из обработки «красных» узлов сетки, обмена между процессами информации о значениях в граничных узлах, обработки «черных» узлов и еще одного обмена значениями граничных узлов.

# 3. Организация экспериментов

Эксперименты проводились на персональном компьютере с процессором Intel Core i7 4700HQ. Для экспериметов использовалась виртуальная машина, которой было выделено 3 Гб оперативной памяти и все 4 ядра процессора. Основной ОС ПК являлась Windows 10, ОС на виртуальной машине являлась Kubuntu 14.04 64-bit. Для виртуализации использовалось ПО VMware Workstation 12.

Для компиляции использовался GCC версии 4.8.4, библиотека mpich версии 3.0.4-6ubuntu1. Основными ключами компиляции были -O3 -DNDEBUG и специфичные для mpich.

Для проведения экспериментов была взята сетка размером 480\*960 узлов. Каждый запуск программы включал в себя 5000 итераций. Результаты экспериментов представлены ниже.

# 4. Результаты экспериментов

На рисунке 2 представлен график ускорения в зависимости от количества процессов для описанной ранее сетки.

Рисунок 2. График ускорения.

На рисунке 3 представлен график эффективности в зависимости от количества процессов для описанной ранее сетки.

Рисунок 3. График эффективности.

Равномерность загрузки процессов вытекает из принудительной синхронизации на каждом шаге итерации. В связи с этим, говорить о ней не имеет смысла.

# 5. Анализ результатов экспериментов и выводы

Итоги экспериментов говорят о том, что алгоритм хорошо поддается распараллеливанию, а выбранная реализация демонстрирует хорошие показатели ускорения и эффективности с ростом числа процессов.

Учитывая линейную зависимость коммуникационной сложности от размера сетки и числа запущенных процессов, данный фактор становится причиной снижения эффективности реализации с ростом их числа.

# 6. Список использованных литературных источников

1. Старченко А.В., Берцун В.Н. Методы параллельных вычислений: Учебник. – Томск: Изд-во Том. ун-та, 2013. – 223 c. – Глава 9. Решение краевых задач для уравнений в частных производных методом конечных разностей.
2. Пазников А.А. Параллельные вычислительные технологии: Практикум. – Новосибирск: Печатный Двор, 2016. – 59 с.