**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

РАБОТА С КЛАСТЕРОМ

Отчет по лабораторной работе № 8 дисциплины

«Высокопроизводительные вычислительные комплексы»

Выполнил студент группы ИВТ-42\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Новокшонов К.Ю./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В.Ю./

Киров 2016

1 Постановка задачи

Необходимо написать программу с использованием MPI для запуска на кластере, протестировать на различном количестве ядер и при различных размерностях данных, построить графики зависимостей времени от количества используемых ядер и от размерности данных.

1.1 Описание схемы Горнера

Схема Горнера — алгоритм вычисления значения многочлена, записанного в виде суммы одночленов, при заданном значении переменной.

Пусть задан многочлен F(x):

(1)

Требуется найти значение функции при x = x0.

Для вычисления значения функции можно разбить выражение 1 следующим образом:

(2)

Для упрощения расчетов необходимо ввести коэффициент b­i, равный:

(3)

Как видно из формулы 3, при вычислениях существует зависимость по данным, то есть следующий шаг вычислений зависит от результата, полученного на предыдущем шаге. Поэтому ускорить вычисления в N раз, где N — количество процессорных элементов, не представляется возможным.

На рисунке 1 показано графическое представление схемы Горнера.

|  |
| --- |
|  |

Рисунок 1 — Графическое представление схемы Горнера

1.2 Постановка индивидуальной задачи

Разработать программу реализации схемы Горнера с использованием средств распараллеливания вычислений MPI.

2 Результаты выполнения

Программа была выполнена на разном количестве блэйдов. Блэйд содержит два четырехядерных процессора Intel Xeon 5345 с частотой 2,33 ГГц и оперативную память объемом 4 Гб. Каждый блейд является системой с общей памятью. На каждом блэйде можно запустить по 8 потоков, соответственно для запуска программы с 16 потоками требуется подключить еще один блэйд.

В ходе выполнения лабораторной работы использовались блэйды с IP-адресами 10.128.1.167, 10.128.1.168, 10.128.1.169.

Результаты выполнения программы вычисления значения многочлена приведены в таблице 1.

График зависимости времени выполнения программы от количества узлов при размерности входных данных, равной 10000000 представлен на рисунке 2.

Таблица 1 — Результаты выполнения программы вычисления значения многочлена

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Количество узлов | 500000 | 750000 | 1000000 | 2500000 | 5000000 | 7500000 | 10000000 |
| 1 | 0,861 | 1,437 | 2,351 | 4,822 | 7,732 | 18,156 | 28,476 |
| 4 | 0,784 | 1,361 | 2,416 | 4,745 | 7,701 | 18,079 | 28,399 |
| 8 | 0,716 | 1,292 | 2,402 | 4,697 | 7,655 | 18,011 | 28,331 |
| 12 | 0,689 | 1,276 | 2,395 | 4,641 | 7,587 | 17,995 | 28,315 |
| 16 | 0,674 | 1,253 | 2,381 | 4,635 | 7,571 | 17,969 | 28,289 |
| 20 | 0,636 | 1,212 | 2,359 | 4,597 | 7,545 | 17,931 | 28,251 |
| 24 | 0,611 | 1,187 | 2,344 | 4,572 | 7,507 | 17,906 | 28,226 |

|  |
| --- |
|  |

Рисунок 2 — График зависимости времени выполнения от количества узлов

График зависимости времени выполнения программы от размерности входных данных при постоянном количестве узлов, равном 8 представлен на рисунке 3.

|  |
| --- |
|  |

Рисунок 3 — График зависимости времени выполнения

от размерности входных данных

График зависимости времени выполнения программы от увеличения количества узлов и размерности входных данных представлен на рисунке 3.

|  |
| --- |
|  |

Рисунок 4 — График зависимости времени выполнения

от увеличения количества узлов и размерности входных данных

3 Выводы

1) Теоретически, при увеличении количества узлов вычисления должны выполняться быстрее. Однако в данной задаче при увеличении количества узлов ускорения не возникает. Связано это с тем, что в задаче существует зависимость по данным, то есть следующий шаг вычислений зависит от результата, полученного на предыдущем шаге, следовательно, данная задача имеет низкую степень параллелизма.

2) При увеличении размерности входных данных время выполнения программы увеличивается. Это оправдано, поскольку вычисления производятся при постоянном количестве узлов.

3) Теоретически, при увеличении размерности входных данных и количества узлов время работы программы должно оставаться постоянным. На графике зависимости времени выполнения программы от увеличения количества узлов и размерности входных данных постоянное время будет выглядеть как прямая линия, параллельная оси X. Однако в данной задаче время увеличивается, что также говорит о низкой степени параллелизма. Это не позволяет получить выигрыш по времени при увеличении количества узлов.