РОЗДІЛ 2. РОЗРОБКА ПРОГРАМИ ПРГ 1 ДЛЯ ПКС СП

2.1 Розробка паралельного математичного алгоритму.

Паралельний математичний алгоритм відповідно до рекомендованої методичної літератури [25] можна подати у вигляді наступних двох етапів:

1. *minZi = min(ZH), i = ;*

*minZ = min(minZ, minZi), i = ;*

1. *MAH = MO∙MKH;*
2. *AH = α∙B∙MAH + minZ∙E∙MRH*,

де:

* ;
* AH – H рядків вектора А;
* MKH – H рядків матриці MK;
* MRH – H рядків матриці MR;
* MAH – H рядків матриці MA

Спільні ресурси: α, B, E, MO

Проаналізуємо розроблений паралельний математичний алгоритм із застосуванням концепції необмеженого паралелізму.

Для оцінки необхідного часу обчислень використаємо теорему Мунро-Петерсона, яка для комп’ютерної системи з необмеженим числом процесорів формулюється так: якщо виконується обчислення скалярної величини, яке потребує ** бінарних операцій, то необхідний час обчислень :

.

Для обчислення minZ = min(Z) необхідно виконати  бінарних операцій порівняння. Тому час виконання буде:

.

2.2 Розробка алгоритмів процесів

Оскільки розроблюване програмне забезпечення є масштабованим і працює на системі із кількістю процесорів P≥2, то зручним варіантом реалізації є написання єдиного алгоритму для всіх задач.

|  |  |
| --- | --- |
| Задачі Т(0) – Т(P-1) | |
| Крок алгоритму | КД, Бар’єри |
| 1. Якщо tid = 0 Ввести α, B, MK, MR. |  |
| 2. Якщо tid = P-1 ввести Z, E, MO. |  |
| 3. **Бар’єр** для усіх задач. Синхронізація по вводу. | Бар’єр |
| 4. Обчислення *minZi = min(Z), i =* |  |
| 5. Обчислення *minZ = min(minZ, minZi), i =* | КД |
| 6. **Бар’єр** для усіх задач. Синхронізація обчислень minZ | Бар’єр |
| 7. **Копіювати** *minZi = minZ, αi = α, Bi = B, Ei = E* | КД |
| 8. Обчислення MAH = MOi∙MKH |  |
| 9. **Бар’єр** для усіх задач. Синхронізація обчислень MA | Бар’єр |
| 10. Обчислення *AH = αi∙Bi∙MAH + minZi∙Ei∙MRH, i =* |  |
| 11. **Бар’єр** для усіх задач. Синхронізація обчислень A | Бар’єр |
| 12. Якщо tid = 0 вивести результат AH. | ­ |

2.3 Розробка схеми взаємодії процесів

На основі алгоритму для всіх задач, наведеному в попередньому розділі, було розроблена структурна схема взаємодії задач (рис. 2.1). Вона дозволяє наочно контролювати бар’єри та критичні ділянки. Крім того, на структурній схемі уводяться також замок, та критична секція, що будуть використовуватись в програмі.

Для демонстрації взаємодії між задачами вибрано три задачі: T(0), T(i), T(P-1) (i = ). Задачі T(0) і T(P-1) вводять дані, тому з ними взаємодіють всі інші (синхронізація по вводу). Задачі T(i) (i = ) виконують лише обчислення і синхронізуються зі всіма іншими задачами по обчисленню. Задача T(0) виводить результат обчислень.



Рис. 2.1 Структурна схема взаємодії задач

2.4 Розробка програми ПРГ1

Програма написана на мові C++ з використанням бібліотеки OpenMP та складається з трьох модулів: основного pro2\_course\_work\_prg1.cpp, та допоміжного operations.cpp і файлу заголовків operations.h.

Основний модуль pro2\_course\_work\_prg1.cpp містить одну функцію:

* main – точка входу в програму. Формує ідентифікатори tid задач, запускає задачі, а також вимірює час виконання elapsedTime програми ПРГ1.

В цьому модулі об’явлена така змінна: P – налаштування кількості процесорів.

Файл заголовків operations.h. представляє заголовки службових функцій (таких як ввід/вивід, копіювання), та типи вектора і матриці. У ньому також об’явлена змінна N що визначає розмірність матриць і векторів.

Допоміжний модуль operations.cpp реалізовує функції об’явлені у файлі заголовків operations.h.

Лістинг програми ПРГ1 наведено у додатку Г.

2.5 Тестування ПРГ1

Для тестування використовувалась паралельна обчислювальна система з наступними апаратними характеристиками:

* процесор
* оперативна пам’ять

В якості програмного забезпечення виступали:

* операційна система:
* середовище розробки і компіляції C++ програми:

Для вимірювання часу виконання програми використовувався високоточний таймер з бібліотеки OpenMP.

Коефіцієнт прискорення  показує скорочення часу виконання паралельної програми в паралельній системі з ** процесорами ** в порівнянні з часом виконання послідовної програми в однопроцесорній системі :



Коефіцієнт ефективності  застосування комп’ютерної системи показує ступінь використання ** процесорів системи:



***Коментар: тут і далі дані будуть отримані та проведені обчислення відповідно ТЗ та плану виконання до 10.05.2015***

Результати тестування і проведених досліджень ефективності розробленої програми наведено в таблицях Таблиця 2.1‑Таблиця 2.3.

Таблиця 2.1 – Час виконання програми ПРГ1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N |  |  |  |  |  |  |
| 900 |  |  |  |  |  |  |
| 1800 |  |  |  |  |  |  |
| 2400 |  |  |  |  |  |  |

На основі даних із таблиці Таблиця 2.1 виконано розрахунок значень коефіцієнтів прискорення, які наведені в таблиці Таблиця 2.2.

Таблиця 2.2 – Коефіцієнти прискорення для програми ПРГ1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Кількість процесорів (P) | | | | | |
| N | 1 | 2 | 3 | 4 | 5 | 6 |
| 900 |  |  |  |  |  |  |
| 1800 |  |  |  |  |  |  |
| 2400 |  |  |  |  |  |  |

Коефіцієнти ефективності (таблиця Таблиця 2.3) обчислено за даними таблиці Таблиця 2.2.

Таблиця 2.3 – Коефіцієнти ефективності для програми ПРГ1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Кількість процесорів (P) | | | | | |
| N | 1 | 2 | 3 | 4 | 5 | 6 |
| 900 |  |  |  |  |  |  |
| 1800 |  |  |  |  |  |  |
| 2400 |  |  |  |  |  |  |

Використовуючи таблиці Таблиця 2.2‑Таблиця 2.3 побудовано графіки зміни коефіцієнтів прискорення і ефективності в залежності від *N* і .

Рисунок 2.2 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N=900

Рисунок 2.3 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N=1800

Рисунок 2.4 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N=2400

Рисунок 2.5 – Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N=900

Рисунок 2.7 – Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N=1800

Рисунок 2.8 – Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N=2400

2.6 Висновки до розділу 2

Виконано розробку програми ПРГ1 для ПКС СП з використанням мови С++ і засобів синхронізації з бібліотеки OpenMP. Тестування програми показало наступне:

* тут будуть результати тестування…