

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни
«Технології паралельних обчислень»

«Застосування високорівневих засобів паралельного програмування для
побудови алгоритмів імітації та дослідження їх ефективності»

Виконав(ла)

ІП-14 Сергієнко Ю. В.
(шифр, прізвище, ім'я, по батькові)

Перевірив

Дифучина О. Ю.
(прізвище, ім'я, по батькові)

Київ 2024

Комп'ютерний практикум 5

Тема: Застосування високорівневих засобів паралельного програмування для побудови алгоритмів імітації та дослідження їх ефективності.

Виконання:

1. З використанням пулу потоків побудувати алгоритм імітації багатоканальної системи масового обслуговування з обмеженою чергою, відтворюючи функціонування кожного каналу обслуговування в окремій підзадачі. Результатом виконання алгоритму є розраховані значення середньої довжини черги та ймовірності відмови. **40 балів.**

2. З використанням багатопоточної технології організувати паралельне виконання прогонів імітаційної моделі СМО для отримання статистично значимої оцінки середньої довжини черги та ймовірності відмови. **20 балів.**

3. Виводити результати імітаційного моделювання (стан моделі та чисельні значення вихідних змінних) в окремому потоці для динамічного відтворення імітації системи. **20 балів.**

Для виконання даних завдань за основу було взято приклад Oracle для Producer-Consumer Application. Для Producer запускається потік, який буде періодично генерувати та відсилати запит до системи. У системі запит завжди поміщається в чергу та дочікується своєї обробки. Якщо один з каналів (Consumer) вільний, він споживає об'єкт з черги або чекає його появи у черзі. Якщо черга заповнена під час запису – відбувається відмова, об'єкт виходить з методу. Для імітації обробки та затримки запитів будемо використовувати випадкові числа, розподілені за нормальним законом та рандомно відповідно.

В окремому потоці було обраховано значення середньої довжини черги та ймовірності відмови: обрахунок здійснювався кожні TIME_LIMIT

/ WAITING_TIME ітерацій при Thread.sleep(WAITING_TIME) для середньостатистичних замірів.

Було створено клас SMOSystem, який в свою чергу запускає декілька потоків класу SMOSubSystem для обрахунку середнього значення rejections та queueSize. Результат обрахунків повертається у вигляді Callable<Values>, де Values – record структура.

```
Processing time: 40000 ms
Iterations: 20
Avg queue: 8.76
Avg rejections: 15.88

Process finished with exit code 0
```

Рисунок 1 – Результати виконання програми

4. Побудувати теоретичні оцінки показників ефективності для одного з алгоритмів практичних завдань 2-5. **20 балів.**

Розрахуємо теоретичні оцінки ефективності алгоритму множення матриць Фокса. Аналізуючи алгоритм, час його виконання займають наступні дії: розділення на частини, їх множення та комбінування результату ($T_1 = n^3$).

Розбиття матриці на блоки займає $m^2 * q^2$, де m – к-сть елементів блоку, q – к-сть блоків по діагоналі. Кількість процесорів визначаємо за q^2 , тому часову складність можна розрахувати наступним чином (1.1).

$$T = m^2 * q^2 = \frac{n^2}{q^2} * q^2 = n^2 \quad (1.1)$$

Комбінування матриці займає скільки ж часу, адже їх сенс однаковий. Щодо алгоритму множення, то його часову складність можна обрахувати за наступною формулою (1.2).

$$T_p = q * m^3 = \frac{n^3}{q^2} = \frac{n^3}{p} \quad (1.2)$$

Часову складність повного алгоритму обрахуємо у формулі 1.3.

$$T_A = \frac{n^3}{p} + 2n^2 = n^2(\frac{1}{p} + 2) \quad (1.3)$$

Прискорення та ефективність алгоритму розрахуємо у формулах 1.4 та 1.5.

$$S_p = \frac{T_1}{T_A} = \frac{n^3}{n^2(\frac{1}{p}+2)} = \frac{n}{\frac{1}{p}+2} \quad (1.4)$$

$$E_p = \frac{T_1}{p * T_A} = \frac{n^3}{p * n^2(\frac{1}{p}+2)} = \frac{n}{1+2p} \quad (1.5)$$

Висновок

Під час виконання даного комп'ютерного практикуму я закріпив знання та навички про використання пулу потоків на прикладі побудови алгоритму імітації СМО системи. Було імітовано роботу декількох споживачів та одного Producer, час роботи яких визначався певним законом розподілу. Також були обраховані оцінки ефективності для алгоритму множення матриць Фокса для поглибленого розуміння покращення паралельного алгоритму над послідовним.

Код програми доступний на [Github](#).