

Sprawozdanie z Projektu i Eksperymentu Obliczeniowego

Laboratorium z Przetwarzania Równoległego

Termin oddania: 30.12.2023

Pierwszy termin oddania: 30.12.2023

Wersja I

Autorzy:

1. Wstęp

Celem niniejszego projektu jest analiza i ocena efektywności przetwarzania równoległego w systemie komputerowym z procesorem wielordzeniowym. Projekt koncentruje się na badaniu wydajności obliczeniowej przy wykorzystaniu pamięci współdzielonej w kontekście zastosowania algorytmów równoległych do rozwiązywania problemu znajdowania liczb pierwszych w określonym przedziale. Kluczowym aspektem jest eksploracja różnych strategii zrównoleglenia, w tym podziału domenowego i funkcyjnego, oraz równoważenie obciążeń procesorów przy użyciu odpowiednio skonfigurowanych wątków OpenMP.

1.2 Opis Wykorzystanego Systemu Obliczeniowego: MacBook Air M2

1. Procesor:

- **Oznaczenie:** Apple M2
- **Liczba procesorów fizycznych:** 1 - integrowany chip Apple M2
- **Liczba procesorów logicznych:** 8 rdzeni CPU (4 rdzenie wydajnościowe, 4 rdzenie efektywności)
- **Liczba rdzeni GPU:** 10 rdzeni GPU

2. Pamięć podręczna procesora:

- **Pamięć L1 i L2:** Zintegrowana, specyfikacja nieopisana przez Apple
- **Pamięć L3:** Zintegrowana, współdzielona, specyfikacja nieopisana przez Apple

3. System operacyjny:

- **Wersja:** MacOS Ventura 13.3

4. Oprogramowanie do kodowania i testów:

- **Nazwa:** Visual Studio Code

5. Dodatkowe specyfikacje:

- **Pamięć RAM:** 8Gb

1.2 Znaczenie i cel ekspreymentu

Projekt ma na celu nie tylko praktyczne zastosowanie teoretycznej wiedzy zdobytej na zajęciach, ale również stanowi okazję do eksploracji nowych technologii i platform sprzętowych. Analiza przeprowadzona na Macbooku Air z procesorem M2 pozwoli na zgłębienie wiedzy na temat możliwości i ograniczeń nowoczesnych technologii Apple w kontekście przetwarzania równoległego.

2. Prezentacja Wariantów Kodów

- **Wariant 1 SP**

```
bool isPrime(int num) {
    if (num <= 1) return false;
    if (num == 2) return true;
    if (num % 2 == 0) return false;

    int limit = std::sqrt(num);
    for (int i = 3; i <= limit; i += 2) {
        if (num % i == 0) return false;
    }
    return true;
}

int countPrimes(int start, int end) {
    std::vector<int> primes;
    for (int i = start; i <= end; i++) {
        if (isPrime(i)) {
            primes.push_back(i);
        }
    }
    return primes.size();
}
```

Ten fragment kodu nie wykonywał przetwarzania równoległego. Wykonywał się sekwencyjnie i służył do weryfikacji wyników i obliczania przyspieszenia.

- **Wariant 2 SS:**

```
std::vector<int> sieveOfEratosthenes(int lower, int upper) {
    std::vector<int> primes;
    std::vector<bool> prime(upper + 1, true);
    prime[0] = prime[1] = false;
    int counter = 0;
    for (int p = 2; p * p <= upper; p++) {
        if (prime[p]) {
            for (int i = std::max(p * p, (lower + p - 1) / p * p); i <=
upper; i += p)
                prime[i] = false;
        }
    }
    for (int p = lower; p <= upper; p++) {
        if (prime[p])
            primes.push_back(p);
    }
    return primes;
}
```

Zakres	Sekwencyjne Klasycznie [SP]	Sekwencyjne Sito [SS]
0 - 1 000 000	0.03869	0.004186
0 - 10 000 000	0.572289	0.036243
0 - 100 000 000	14.0072	0.37441

Opis Przebiegu Przetwarzania (dla poszczególnych wariantów kodu)

- **Podział Pracy:** [Wielkość zbioru zadań], [Sposób przydziału zadań do procesów].
- **Dyrektywy OpenMP:** [Wykorzystane dyrektywy i ich znaczenie].
- **Problemy Poprawnościowe:** [Omówienie wyścigów i ich wpływu].
- **Problemy Efektywnościowe:**
 - **False Sharing:** [Wyjaśnienie false sharing i jego wpływu].
 - **Synchronizacja:** [Czy, gdzie i jak synchronizacja wpływa na czas obliczeń].

Punkt 3: Prezentacja Wyników i Omówienie Eksperymentu

- Testowane Wersje Kodów:** [Wariant 1, Wariant 2, ...].
- Tabela Wyników:**
- Omówienie Wyników:**

Porównanie Jakości Rozwiązań: [Prędkość przetwarzania vs. czas obliczeń].

Analiza Efektywności Zrównoleglenia: [Przyspieszenie, Efektywność, Prędkość].

Punkt 4: Wnioski

Porównanie Podejść: [Wariant 1 vs. Wariant 2, ...].

Podsumowanie Zrównoważenia Przetwarzania: [Analiza zrównoważenia pracy procesorów].

Ocena Efektywności Skalowania: [Efektywność w zależności od liczby procesorów].

Ograniczenia Efektywnościowe: [Dominujące ograniczenia w kodzie].

(Używanie miar względnych: np. "czas przetwarzania 2 razy krócej" zamiast "o 2 sekundy krócej").

Przygotowanie Sprawozdania

Format: Plik PDF z numeracją stron.

Dodatki: Plik ZIP z kodami źródłowymi.

Numeracja: Numeracja stron i obiektów w sprawozdaniu.

Termin: Do 30.12.2022.

Podsumowanie i Ocena Realizacji Projektu

Po sprawdzeniu merytorycznej zawartości sprawozdania, możliwa jest rozmowa z autorami. Opóźnienia obniżają ocenę.