

ПОРІВНЯННЯ МЕТОДІВ РОЗРОБКИ паралельних програм

Мета заняття

Отримати практичні навички розробки паралельних програм та порівняти різні методи (використання засобів операційної системи та спеціальних засобів їх розробки)

Методичні вказівки до організації самостійної роботи

1. Освоїти етапи розробки паралельних програм [1, с. 159-176].
2. Вивчити критерії оцінки паралельних програм [1, с. 49-86].
3. Повторити технологію створення потоків в Windows (функція CreateThread) [2, с. 160-1776, с. 187-245].
4. Вивчити директиви Open MP для створення паралельних програм [4, с. 265-346, 5, с. 3-52].
5. Дослідити методи обчислення числа π .

Методи обчислення числа π

Доказано, що нема точного значення π . Його можна обчислити лише приблизно. Є дуже багато методів обчислення числа π (По запиту Число π Google видає сотні тисяч посилань). Розглянемо декілька з них.

1 Зі школи знаємо, що довжина кола дорівнює $l = 2\pi r$. Якщо його радіус $r = 1$, то $l = 2\pi$. Впишемо в коло правильний багатокутник з n сторонами. Тоді його периметр p при збільшенні n буде приблизно дорівнює 2π . Недолік – треба використовувати функцію $\sin(\alpha/2)$, де $\alpha = 360/n$, а функція $\sin$ обчислюється довго.

2 Зобразимо коло на координатній площі з центром на початку координат. Площа круга дорівнює $s = \pi r^2$. Площа частини круга в першій чверті $s = \pi r^2/4$. Для $r = 1$ маємо $s = \pi/4$. З іншого боку з урахуванням рівняння кола для центра на початку координат і $r = 1$ маємо $x^2 + y^2 = 1$:

$$s = \int_0^1 \sqrt{1-x^2} dx$$

Для обчислення інтегралу можна використовувати приблизні методи прямокутника, трапеції,...

3 Для паралельних обчислень найчастіше використовується метод обчислення:

$$\int_0^1 \frac{4}{1+x^2} dx,$$

Який отримуємо так:

$$\pi/4 = \arctg 1 = \int_0^1 \frac{1}{1+x^2} dx$$

Для обчислення інтегралу можна використовувати приблизні методи прямокутника, трапеції,...

4 Використання випадкових чисел. На координатній площі зобразимо квадрат з центром на початку координат і стороною, рівною 2. Впишемо в цей квадрат коло. Обчислимо площу квадрату в першій чверті S_1 , вона дорівнює $S_1 = 1$. Обчислимо площу частини круга в першій чверті S_2 , $S_2 = \pi/4$. Тоді $S_2/S_1 = \pi/4$. Будемо генерувати напів випадкові числа в діапазоні $x = 0 \dots 1$; $y = 0 \dots 1$. Будемо обчислювати загальну кількість крапок, та кількість крапок, які в середині круга ($x^2 + y^2 \leq 1$). Згідно методу Монте – Карло відношення цих кількостей приблизно дорівнює відношенню площин і тим точніше, чим більше загальна кількість крапок і чим генератор ближче до рівномірно розподіленого.

5 Ряд Лейбниця :

$$\pi/4 = 1/1 - 1/3 + 1/5 - 1/7 + \dots$$

Контрольні запитання і завдання

1. Чим відрізняються етапи розробки паралельних програм від етапів розробки послідовних програм?
2. Які критерії оцінки паралельних програм Ви знаєте?
3. Що визначає і для чого використовується закон Амдаля?
4. Яким чином можна визначити потокову функцію, якщо їй необхідно передати декілька параметрів?
5. Які директиви використовуються для завдання паралельної частини коду (Open MP)?
6. Коли треба задавати секції паралельного коду (Open MP)?
7. Як розділяються змінні в Open MP в залежності від області їх відомості?

Приклади аудиторних задач

1. Скласти функції для обчислення числа π за допомогою 3 – 5 способів. Обрати найбільш ефективний з боку часу обчислення для досягнення однакової точності.

```
// 3
int n = 10000000;
double sCalcPi3 (){
int i;
    double h = 1. / n , pi = 0;
    for(i=0; i < n; i++){
        double x = h * i;
        pi += (4. / (1 + x * x)) ;
    }
    pi = h * pi;
    return pi;
}

// 4
```

```

#define RAND01() (((rand () % 4096))/4096.)
double sCalcPi4 (){
    double x, y;
    int m = 0;
    for (int i = 0; i < n; i++){
        x = RAND01();
        y = RAND01();
        if (x * x + y * y <= 1) m++;
    }
    return 4. * m / n;
}
// 5
double sCalcPi5 (){
    double s = 0;
    for (int i = 1; i < n; i += 4){
        s += 1./ i;
        s -= 1./ (i + 2);
    }
    return 4. * s;
}

```

Зробити висновки по бистроті методів.

2. Скласти потокову функцію для обчислення значення інтегралу в заданому діапазоні (для найкращого методу – ми тут показуємо для 3 методу)

```

typedef struct{
    double h, s;// s - результат
    int begin, end, step;
} THREADSDATA, *PTHREADSDATA;

DWORD WINAPI ThreadFun (PVOID pData){
    PTHREADSDATA p = (PTHREADSDATA)pData;
    double s = 0, x, h = p->h ;
    for (int i = p->begin; i < p->end; i+= p->step){
        x = i * h;
        s += 4 / (1 + x * x);
    }
    p->s = s * h;
    return 0;
}

```

1. Перейти в режим використання Open MP та визначити кількість потоків.

```

#include <omp.h>
...
#ifdef _OPENMP
    cout << "Open MP support\n";
#else
    cout << "Open MP not support\n";
#endif
...
int count = omp_get_max_threads ();

```

3. Використовуючи потоки Windows виконати обчислення числа π .

```

double pThreadCalcPi3 (){
    int nThreads = omp_get_max_threads ();
    double h = 1./n;
    HANDLE *hThread = new HANDLE [nThreads - 1];
    PTHREADSDATA pDataas = new THREADSDATA [nThreads];
    int delta = n / nThreads;
    for (int i = 0; i < nThreads; i++)
    {
        pDataas [i].begin = i * delta ;
        pDataas [i].end = pDataas [i].begin + delta;
        pDataas [i].step = 1;
        pDataas [i].h = h;
        if (i != nThreads - 1)
            hThread [i] = CreateThread (0, 0, ThreadFun, &pDataas [i], 0, 0);
    }
    ThreadFun (&pDataas [nThreads - 1]);
    WaitForMultipleObjects (nThreads - 1, hThread, TRUE, INFINITE);
    double pi = 0;
    for (int i = 0; i < nThreads - 1; i++)
    {
        CloseHandle (hThread [i]);
        pi+= pDataas [i].s;
    }
    pi+= pDataas [nThreads - 1].s;
    return pi;
}

```

4. Реалізувати функцію обчислення числа π за допомогою Open MP

```

double pCalcPi3 (){
    double h = 1. / n , pi = 0;
#pragma omp parallel for reduction (+ : pi)
    for(int i=0; i < n; i++)
    {
        double x = h * i;
        pi += 4. / (1 + x * x) ;
    }
    pi = h * pi;
    return pi;
}

```

5. Визначити обчислювальну складність функції.

6. Порівняти характеристики паралельних програм.

Приклади домашніх задач

1. Обчислити число π за допомогою ряду Лейбница і порівняти 2 метода обчислення.
2. Скласти функції для обчислення додатку матриць в послідовному режимі.
Використати звичайний та оптимізований режим.

3. Скласти функцію для обчислення додатку матриць для найбільш ефективного алгоритму (паралельний режим).