Технології паралельного програмування

Тема: Труднощі паралельного і розподіленого програмування

Лекція №6

Проф. Н.Г. Аксак

кафедра КІТС, ХНУРЕ

План лекції

1 Паралелізм

2 Фундаментальні поняття паралельного програмування

3 Моделі розподіленого програмування

4 Застосування паралелізму

5 Деякі особливості розпаралелювання

Характеристики традиційних алгоритмів:

- число операцій,
- обсяг необхідної пам'яті
- точність.

теорія процесів

Теорія процесів є одним з розділів математичної теорії програмування, який вивчає математичні моделі поведінки динамічних систем, звані процесами**.**

Процес являє собою модель поведінки, яка полягає у виконанні дій: прийом або передача будь-яких об'єктів або перетворення цих об'єктів**.**

process calculus -процесна алгебра

π-calculus- пі-обчислення основна модель поведінки мобільних взаємодіючих систем

CCS (Calculus of Communicating Systems) - Обчислення взаємодіючих систем

А.М.МироновА.М.Миронов Теория процессовА.М.Миронов Теория процессов
http://intsys.msu.ru/staff/mironov/processes.pdf

Переваги теорії процесів

Для формального опису та аналізу поведінки розподілених динамічних систем:

1. компоненти працюють паралельно,
2. взаємодія компонентів відбувається шляхом пересилки сигналів або повідомлень від одних компонентів іншим компонентам:
3. Дозволяють аналізувати з прийнятною складністю моделі з дуже великим і навіть нескінченною безліччю станів.
4. Добре підходять для вивчення ієрархічних систем, тобто таких систем, які мають багаторівневу структуру.

Верифікація процесів

Верифікація - побудова формального доказу того, що аналізований процес має задані властивості.

Наприклад, безпечна експлуатація таких систем, як

- системи управління атомними електростанціями,
- медичні пристрої з комп'ютерним управлінням,
- бортові системи управління літаків і космічних апаратів,
- системи управління секретними базами даних,
- системи електронної комерції

Постановка завдання верифікації

1. Побудова процесу, що представляє собою математичну модель поведінки аналізованої системи.

2. Подання властивості, що перевіряється у вигляді математичного об'єкта (званого специфікацією).

3. Побудова математичного доведення твердження про те, що побудований процес задовольняє специфікації

Паралелізм

Паралельна обробка має два різновиди:

- конвеєрність
- паралельність

Для написання паралельної програми необхідно

- Виділити групи операцій, які можуть обчислюватися одночасно
- Виявити відсутність в програмі справжніх інформаційних залежностей

Дві операції програми називаються інформаційно залежними, якщо результат виконання однієї операції використовується в якості аргументу в інший.

Неформальне поняття процесу

Процес являє собою граф P, де

- Вершини графа P називаються станами, і зображують ситуації, в яких може перебувати система , що моделюється в різні моменти свого функціонування.

Одне з станів є виділеним, воно називається початковим станом процесу P.

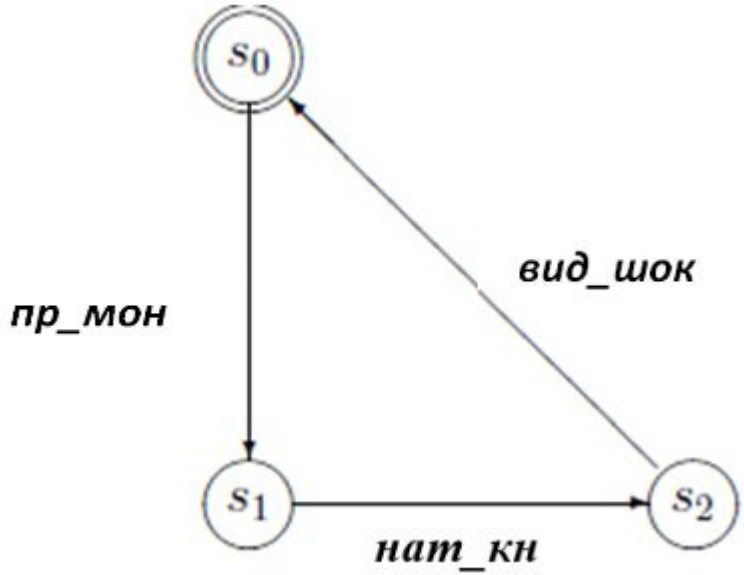
- Ребра графа P мають мітки, що зображують дії, які може виконувати система , що моделюється.
- Функціонування процесу P описується переходами по ребрах графа P від одного стану до іншого. Функціонування починається з початкового стану.

Мітка кожного ребра зображує дію процесу, який виконувався під час переходу від стану на початку ребра до стану в його кінці.

Приклад: процес, який являє собою найпростішу модель поведінки торгового автомата.

Позначимо дії короткими іменами:

- прийом монети ми позначимо символом ***пр_мон***,
- натискання кнопки - символом ***нат_кн***,
- видачу шоколадки - символом ***вид_шок***.



Визначення поняття процесу

Процесом називається трійка P виду

$$P = (S, s^0, R)$$

компоненти якої мають наступний сенс.

- **S** - множина, елементи якої називаються станами процесу P
- $\mathbf{s^0 \in S}$ – деякий виділений стан, званий початковим станом процесу P.
- R – підмножина виду

$$\mathbf{R \subseteq S \times Act \times S}$$

Елементи множини R називаються переходами.

Якщо перехід з R має вигляд $\mathbf{(s^1, a, s^2)}$, то

– будемо говорити, що цей перехід є переходом зі стану $\mathbf{s^1}$ в стан $\mathbf{s^2}$ з виконанням дії **a**,

– стани $\mathbf{s^1}$ и $\mathbf{s^2}$ називаються початком і кінцем цього переходу відповідно, дія ***a*** називається міткою цього переходу

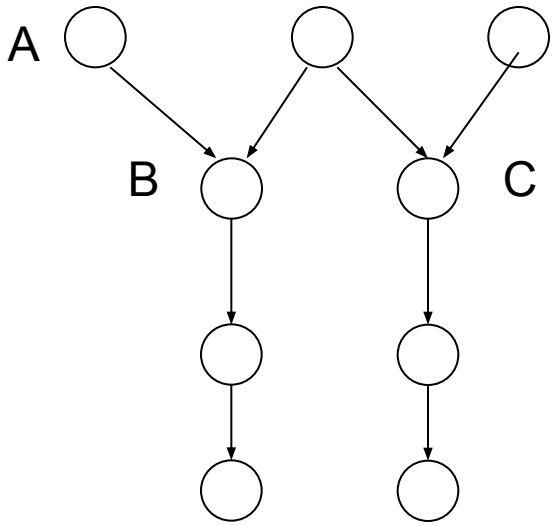
Паралелізм

Завдання розпаралелювання:

Знаходження інформаційно незалежних операцій

Розподіл операцій між обчислювачами

Забезпечення синхронізації



граф інформаційних залежностей

Фундаментальні поняття паралельного програмування

Декомпозиція - розбивка програми на індивідуальні підзадачі і ідентифікація залежностей між ними

Основні форми декомпозиції

Декомпозиція	Опис	коментар
Задача	Різні дії призначені для різних потоків	Часто зустрічається в GUI додатках
Дані	Множина потоків виконують одну і ту ж операцію над різними блоками даних	Часто зустрічається в обробці звуку, зображень і в наукових розробках
Потік даних	Вихідні дані одного потоку є вхідними для іншого	Необхідно приділити увагу на усунення початковій і кінцевій затримок

Паралелізм

Завдання декомпозиції - розкладання програми на функції

Паралелізм рівня даних, розбиває завдання за умовою їх впливу на дані

Декомпозиція потоків породжує проблему розподілу потоку даних між завданнями

Паралелізм

Деякі можливі проблеми:

- *Синхронізація*
- *комунікація*
- *балансування завантаження*
- *масштабованість*

Модели параллельного программирования

Шаблон	Декомпозиція
Паралелізм рівня завдання	Задача
Розділяй і володарюй	Задача/дані
Геометрична декомпозиція	Дані
Конвейер	Потік даних
Хвильова обробка даних	Потік даних

CCS (Calculus of Communicating Systems).

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8
5	6	7	8	9

Моделі розподіленого програмування

- "Клієнт / сервер"
- "Виробник-споживач"
- мультиагентні розподілені системи

π-calculus

Використання паралелізму

Способи розпаралелювання:

- **великоблочне розпаралелювання**
- **розподіл ітерацій циклів**
- **блоковий розподіл**
- **циклічний розподіл**

Використання паралелізму

Великоблочне розпаралелювання:

розподілу робіт між процесорами

{/ * Операції, що виконуються 0-м процесором */}

(if (Myproc==0)

….

{/* Операції, що виконуються k-м процесором */}

if (Myproc==K)

….

Використання паралелізму

Розподіл ітерацій циклів.

```
for (i = 0; i < N; i++)
 {
      if (i ~ MyProc)
         {
/* операції і-ї ітерації для виконання процесором Myproc*/
   }
  }
```

Використання паралелізму

Блоковий розподіл ітерацій

Кількість ітерацій циклу N ділиться на число процесорів p, результат округляється до найближчого цілого зверху і число N / p визначає кількість ітерацій в блоці..

Використання паралелізмузма

```
for (i = 0; i < N; i++)
  a[i] = a[i] + b[i];
```

Використання паралелізму

Блоковий розподіл ітерацій

```
k=(N-1)/p+1; /* розмір блоку иітераціий */
ibeg=MyProc*k;  /* початок блоку ітерацій процесора MyProc*/
iend=(MyProc+1)*k-1;  /* кінець блоку ітерацій MyProc */
if (ibeg>=N) iend=ibeg-1; /* якщо процесору не дісталося ітерацій */
else if (iend>=N) iend=N-1; /* якщо процесору дісталося менше ітерацій */
for (i=ibeg; i<=iend; i++)
    a[i]=a[i]+b[i];
```

Використання паралелізму

- **циклічний розподіл** :

```
for (i = MyProc; i < N; i+=P)
    a[i] = a[i] + b[i];
```

Використання паралелізму

1 математична постановка задачі, запис в формульному вигляді;

2 побудова обчислювального алгоритму;

3 створення і оптимізація послідовної програми;

4 дослідження ресурсу паралелізму програми і вибір методу розпаралелювання;

5 розробка паралельного алгоритму

6 рівномірно завантажити процесори

7 мінімізувати кількість і обсяг даних, що

Використання паралелізму

Багатовимірні циклічні гнізда

Простором ітерацій гнізда тісно вкладених циклів називають множинУ цілочисельних векторів I, координати яких задаються значеннями параметрів циклів даного гнізда.

Завдання розпаралелювання при цьому зводиться до розбиття множини векторів I на підмножини, які виконуються послідовно один за одним, але в рамках кожної такої підмножини ітерації можуть бути виконані одночасно і незалежно.

Використання паралелізму

Методи аналізу простору ітерацій:

координат

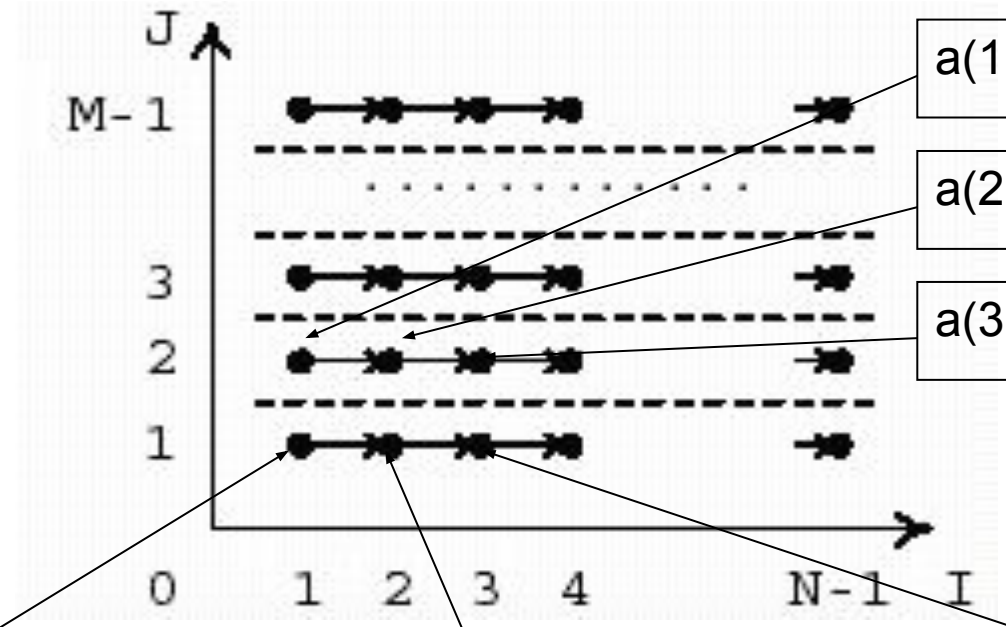
гіперплоскостей

паралелепіпедів

пірамід

Приклад (метод координат)

```
for (i = 1; i < N; i++)
   for (j = 1; j < M; j++)
    a[i,j] = a[i-1,j] + a[i,j];
```



a(1,2)= a(0,2)+ a(1,2)

a(2,2)= a(1,2)+ a(2,2)

a(3,2)= a(2,2)+ a(3,2)

a(1,1)= a(0,1)+ a(1,1)

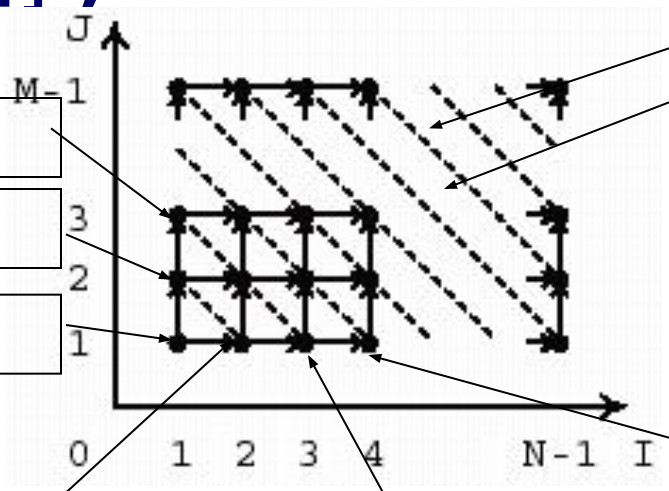
a(2,1)= a(1,1)+ a(2,1)

a(3,1)= a(2,1)+ a(3,1)

Приклад 2

метод гіперплоскостей

```
for (i = 1; i < N; i++)
  for (j = 1; j < M; j++)
    a[i,j] = a[i-1,j] +
  a[i,j-1];
```



Гіперплоскості $i+j=const$

a(1,3)= a(0,3)+ a(1,2)

a(1,2)= a(0,2)+ a(1,1)

a(1,1)= a(0,1)+ a(1,0)

a(2,1)= a(1,1)+ a(2,0)

a(2,2)= a(1,2)+ a(2,1)

a(2,3)= a(1,3)+ a(2,2)

Приклад 3

```
for (i = 1; i < N; i++)
{
  a[i] = a[i] + b[i];                    (3.1)
  c[i] = c[i-1] + b[i];
}
```

"

```
for (i = 1; i < N; i++)

a[i] = a[i] + b[i];

for (i = 1; i < N; i++)                  (3.2)
```

Контрольні питання

1. До чого зводиться задача розпаралелювання програми?
2. У яких випадках можливе ефективно розпаралелити програму?
3. Що необхідно зробити, для того щоб змусити паралельну обчислювальну систему або суперЕОМ працювати з максимальною ефективністю на конкретній програмі?
4. Перерахуйте паралельні шаблони програмування та зробіть короткий огляд типів проблем, до яких може бути застосований кожен зразок.
5. Розробіть граф "операції-операнди« для прикладу (3.2).
6. Розкрийте послідовність дій при створенні паралельної програми, що реалізує множення двох векторів.