

Подход за конструиране на паралелни алгоритми

Гергана Бенчева

ИПОИ – БАН

gery@parallel.bas.bg

Съдържание

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

- Паралелна обработка на информация
- Подход за конструиране на ПА
- Оценка на ефективността
- Средства за реализация

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

Паралелна обработка на информация

Паралелна обработка на информация

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

Един пример: За колко време могат да се измият една/няколко коли, ако 1 човек измива 1 кола за 1 час?

- 2 души измиват 2 коли за 1 час?
- 2 души измиват 1 кола за $1/2$ час? Как си разделят работата?
- 10 души измиват 10 коли за 1 час?
- 10 души измиват 1 кола за $1/10$ час?

Видове паралелна обработка:

- ”в дълбочина”
- ”на ширина”
- ”реална”
- ”симулирана”

Паралелни компютри

Паралелна
обработка

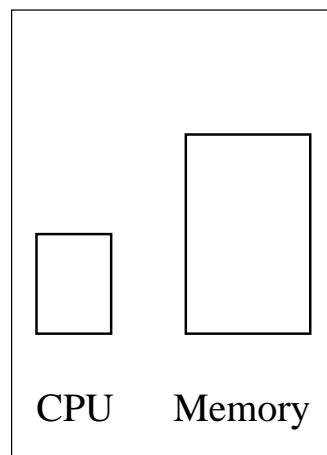
Конструиране

Ефективност

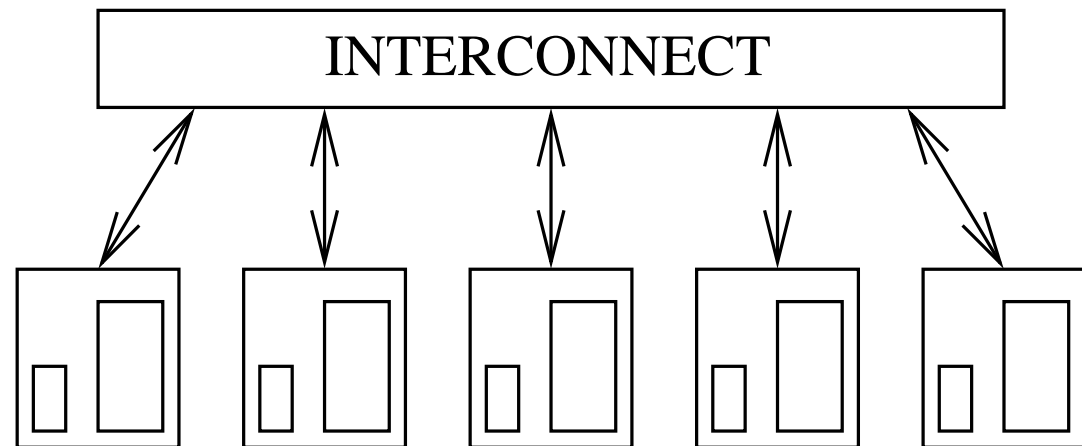
Средства за
реализация

Паралелен компютър – множество процесори, които могат да работят съвместно за решаване на изчислителна задача.

Модел на паралелна машина



Фон Нойман



Мултикомпютър

MIMD с разпределена памет

MIMD с обща памет

SIMD (векторни компютри)

Паралелна
обработка

Конструирание

- Основни етапи
- Разделяне
- Комуникации
- Агломерация
- Изобразяване

Ефективност

Средства за
реализация

Как да конструираме ПА?

Основни етапи

Паралелна
обработка

Конструиране

● Основни етапи

● Разделяне

● Комуникации

● Агломерация

● Изобразяване

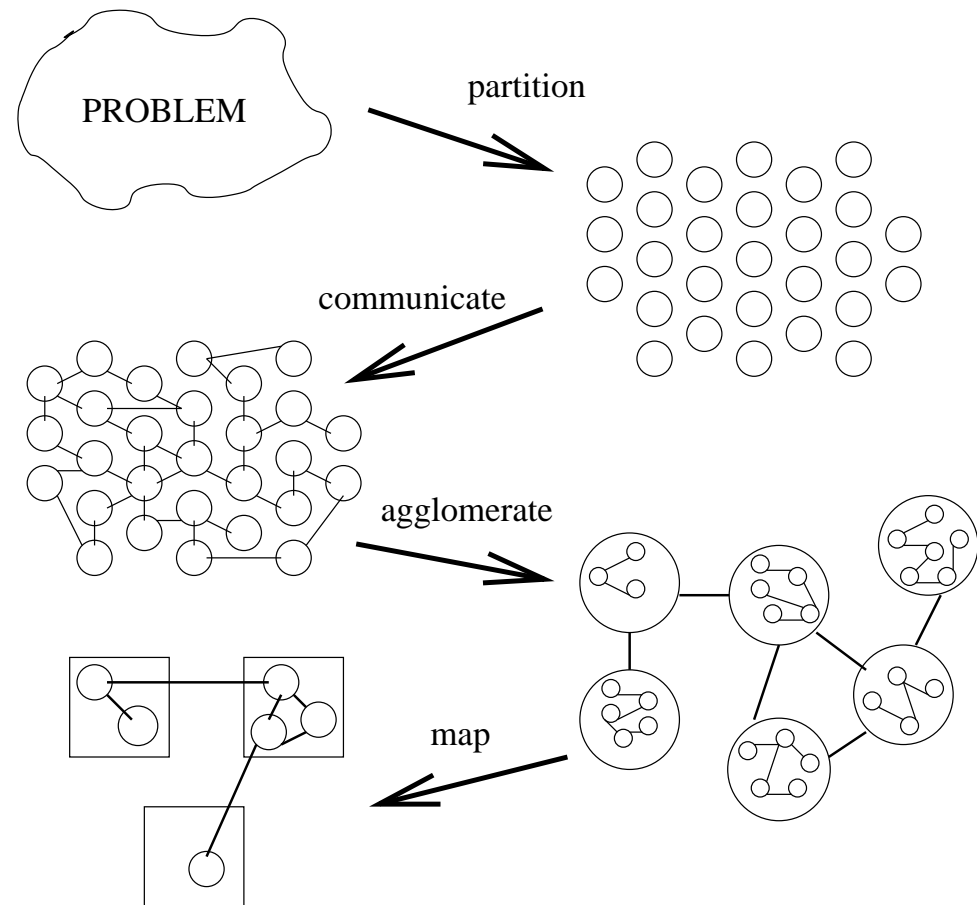
Ефективност

Средства за
реализация

1. Разделяне
2. Комуникации
3. Агломерация
4. Изобразяване

1 и 2 – не зависят
от машината

3 и 4 – конкрет-
на паралелна архи-
тектура и произво-
дителност



I. Foster, *Designing and building parallel programs*, Addison-Wessley, 1995.

Разделяне

Паралелна
обработка

Конструиране

- Основни етапи
- **Разделяне**
- Комуникации
- Агломерация
- Изобразяване

Ефективност

Средства за
реализация

Изчисленията, които трябва да се извършат и данните, с които се оперира се разделят на малки задания (по възможност – непресичащи се). Практически въпроси, като брой процесори в достъпния компютър се игнорират и вниманието се насочва върху откриване на възможности за паралелно изпълнение.

а) **разделяне на областта** (domain decomposition) – първо се определя подходящо разделяне на данните на задачата и след това се свързват изчисленията с данните.

б) **функционално разделяне** (functional decomposition) – първо се разделят изчисленията и след това се разглеждат данните.

Списък за проверка

1. Дефинирани ли са поне с порядък повече задания отколкото са процесорите на достъпния компютър?
Ако не – имате по-малка гъвкавост в следващите етапи на схемата.
2. Избягват ли се излишни пресмятания и място за съхранение на данни?
Ако не – крайният алгоритъм може да не е достатъчно ефективен за големи задачи.
3. Задчите със сравнима големина ли са?
Ако не – може да е трудно да се осигури равно количество работа за всички процесори.
4. Зависи ли броят на подзаданията от размерността на задачата? В идеалния случай увеличаването на размерността на задачата води до увеличаване на броя на заданията, а не на тяхната размерност.
Ако не – ПА няма да може да решава по-големи задачи, когато са достъпни повече процесори.
5. Определени ли са няколко различни разделяния? Дават възможност за увеличаване на гъвкавостта на следващите етапи.

Комуникации

Паралелна
обработка

Конструиране

- Основни етапи
- Разделяне
- **Комуникации**
- Агломерация
- Изобразяване

Ефективност

Средства за
реализация

Определят се необходимите комуникации за съгласуване на изпълнението на заданията и се дефинират подходящи комуникационни структури. Целта е да се разпределят операциите за комуникация (без да се въвеждат ненужни такива) върху много задания и да се организират по начин, който позволява едновременно изпълнение.

- **локални/глобални** – всяко задание комуникира с малко/голямо множество други задания;
- **структурирани/неструктурирани** – задание и неговите съседи образуват регулярна структура (напр. дърво или мрежа)/произволен граф;
- **статични/динамични** – вида на комуникационните структури не се променя с времето или се определя според изчислените данни на отделни етапи от изпълнението на алгоритъма;
- **синхронни/асинхронни** – производителят и потребителят на данните си съдействат/не си съдействат в операциите за обмен.

Списък за проверка

1. Дали всички задания изпълняват приблизително равен брой комуникации?

Ако не – неефективна конструкция. Преразгледайте възможността за по-равномерно разпределение на операциите за комуникация. Напр., ако структура от данни, до която често има достъп се намира в едно задание, разгледайте възможността за нейното разделяне или дублиране.

2. Дали всяко задание комуникира с малък брой съседни?

Ако не – оценете възможността тази глобална комуникация да се преформулира в термините на локална комуникационна структура.

3. Възможно ли е операциите за комуникация да се извършват едновременно?

Ако не – вашият алгоритъм изглежда е неефективен и без ускорение. Използвайте подхода ”разделяй и владей” (”divide-and-conquer”)

4. Възможно ли е изчисленията свързани с различни задания да се извършват едновременно?

Ако не – вашият алгоритъм изглежда е неефективен и без ускорение. Възможно ли е пренареждане на операциите за комуникации и изчисления?

Агломерация

Заданията и комуникационните структури от първите два етапа се оценяват по отношение на конкретната производителност и реализацията. Целта е да се получи ефективен алгоритъм за определен клас паралелни компютри. Ако е необходимо, заданията се групират в по-големи и се дублират данни и/или изчисления, за да се подобри производителността или за да се намали цената за разработване.

Списък за проверка

1. Дали агломерацията намалява цената на комуникациите, чрез увеличаване на локалността?
2. Дали ползата от евентуално дублиране на изчисления е по-голяма от цената му за множество различни размерности на задачата и брой процесори?
3. Дали евентуалното дублиране на данни не излага на риск ефективността чрез намаляване на диапазона на размерностите на задачите или броя на процесорите?
4. Агломерацията доведе ли до задания с приблизително равностойни цени на изчисленията и комуникациите?
5. Дали броя на заданията все още зависи от размерността на задачата?

- Основни етапи
- Разделяне
- Комуникации
- Агломерация
- Изобразяване

Всяко задание се свързва с процесор по такъв начин, че да се удовлетворят конкуриращите се условия за максимално използване на процесорите и минимизиране на комуникациите. Изобразяването може да бъде зададено статично или да се определя по време на изпълнението чрез алгоритми за балансиране на натоварването. (Върху системи с обща памет това се прави автоматично.)

Баланс между две стратегии:

1. Увеличаване на паралелизма – задания, които могат да се изпълняват едновременно се разполагат върху различни процесори.
2. Увеличаване на локалността – задания, които су комуникират често се разполагат в един и същ процесор.

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

ЕФЕКТИВНОСТ

Оценка на ефективността

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

$$\text{Ускорение: } S_P = \frac{T_1}{T_P} \quad S_P \leq P$$

$$\text{Ефективност: } E_P = \frac{S_P}{P} = \frac{T_1}{P \cdot T_P} \quad E_P \leq 1$$

MIMD с разпределена памет и P процесора

$$T_P = T_a + T_{com}, \quad T_a = M * t_a, \quad T_{com} = t_s + l M * t_w$$

t_a е усреднено време за извършване на 1 ар. оп. от 1 процесор

t_s е времето за стартиране на комуникация

t_w е времето за изпращане на едно число до съседен процесор

$$T_P^{Alg} = \mathcal{N}^{Alg} * t_a + c_1 * t_s + c_2 * t_w$$

I. Foster, Designing and building parallel programs, Addison-Wesley, 1995.

V. Kumar, A. Grama, A. Gupta, G. Karypis, Introduction to parallel computing: design and analysis of algorithms, Addison-Wesley, 1994.

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

Средства за реализация

Средства за реализация

Паралелна
обработка

Конструиране

Ефективност

Средства за
реализация

Програмиране чрез предаване на съобщения: Паралелна система от няколко процеса, които имат тяхна локална памет и могат да комуникират помежду си чрез съгласувани трансфери на данни от един процес до друг.

MPI (Message Passing Interface) преносим стандарт за предаване на съобщения.

- включва реализации на Fortran77 и C/C++
- достъпен върху UNIX, Linux, Windows и др.
- скрива разликите в компютърните архитектури

MPI дефинира

ком. м/у 2 процесора

групови комуникации

групи от процеси

комуник. области

MPI не дефинира

явни операции с обща памет

средства за констр. на програми

средства за откриване на грешки

входно/изходни функции (верс. 1.x)

Standard: <http://www.mpi-forum.org>

MPICH: www.mcs.anl.gov/research/projects/mpi/mpich2