

Комп'ютерне моделювання задач прикладної математики

Технології паралельних обчислень.

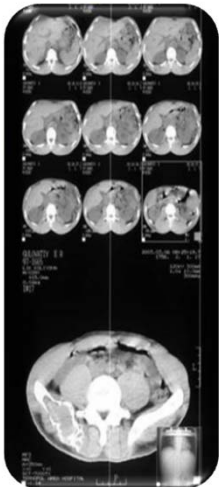
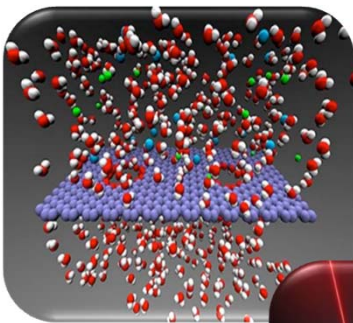
Лекція 4

**Технологія
паралельних
обчислень актуальна,
бо...**

Є засобом розв'язання
задач, які знаходяться
на передовому фронті
сучасної науки.

Легкодоступна для
досліджень: досить мати
базові знання, щоб можна
було розвиватись самому і
розвивати галузь

Є дуже широкою сферою
діяльності: розробка числових
методів, вивчення структурних
властивостей алгоритмів,
створення нових мов
програмування, конструювання
унікальних технічних рішень тощо.



Для яких задач є зміст застосовувати паралельні обчислення?

Там, де треба забезпечити:

- ❖ надвисоку швидкодію,
- ❖ великий об'єм оперативної пам'яті,
- ❖ велику кількість інформації, що передається,
- ❖ обробку і зберігання великого об'єму інформації,
- ❖ висока відмовостійкість.

Актуальні напрямки, в яких необхідна паралелізація процесів:

- 1) чисельне розв'язання задач великої розмірності із багатьма змінними;
- 2) моделювання об'єктів і проведення аналізу поведінки складних систем різної природи;
- 3) управління складними промисловими та технологічними процесами в режимі реального часу і в умовах невизначеності;
- 4) обробка великих об'ємів інформації.

Сучасні задачі обчислювальних систем:

- генна інженерія,
- створення лікарських препаратів,
- розрахунки в квантовій фізиці, хімії,
- обробка даних прямих спостережень в астрофізиці,
- моделювання клімату,
- криптографія та шифрування,
- моделювання економічних систем,
- он-лайн ігри,
- інтелектуальний аналіз даних,
- соціальні мережі,
- потік запитів в пошукових або клієнт-серверних системах та ін.
- розрахунок та проектування електронних структур,
- моделювання і прогнозування соціальних процесів,
- 3D анімація, рендерінг, обробка відео та аудіо.

Основна мета паралельних обчислень – зменшення часу рішення задачі (у випадку рішень в реальному часі або при необхідності дуже великого обсягу обчислень).

Ще:

- Забезпечити при розв'язання великих задач:
 - збільшення деталізації;
 - підвищення точності.
- Організувати велике інформаційне навантаження:
 - розподілити вхідний потік запитів;
 - розподілити місце збереження даних.
- Використовувати усі доступні вільні ресурси.

Задача паралельних обчислень - створення ресурсу паралелізму (одержання паралельного алгоритму) в процесах вирішення завдань і управління реалізацією цього паралелізму з метою досягнення найбільшої ефективності використання відповідної обчислювальної техніки.

1. **Забезпечення надвисокої швидкодії** – складні, багатовимірні задачі, які необхідно розв'язати на протязі досить обмеженого часу.
Приклад: задачі прогнозу погоди.

Прогноз погоди з передбачуваністю **10 днів з 10-ти хвилинним кроком:**

- продуктивність системи 100 Mflops – витрачений час 10^7 секунд (~ 100 дн.).
- продуктивність 1.7 Tflops – витрачений час 10 хв.

gismeteo прогноз погоди

Кожаная обувь по крутой цене
 Распродажа началась, успеи на скидку

Головна Карти **Погодні новини** Прогноз на місяць Інформери Додатки Пош

Погода
 Київ
 Київ, Україна

+26°C 3 м/с ПнС 752 мм рт. ст. 49% волог. 20°C вода

Ясно

1 вересня 2018 18:00

У Києві

- Стилий прогноз погоди
- Погодинний прогноз погоди
- Погода на 2 тижні
- Погода на місяць
- Геоманітний стан
- Погода по-старому
- Погода для зайнятих
- Щоденник погоди

Астрономія

Сонце Місяць

Схід 6:11
 Захід 19:43
 Тривал. 13:31

Фаза 65%
 Старіючий

Прогноз Сб-Пн Пн-Ср Ср-Пт Пт-Нд 2 тижні Місяць

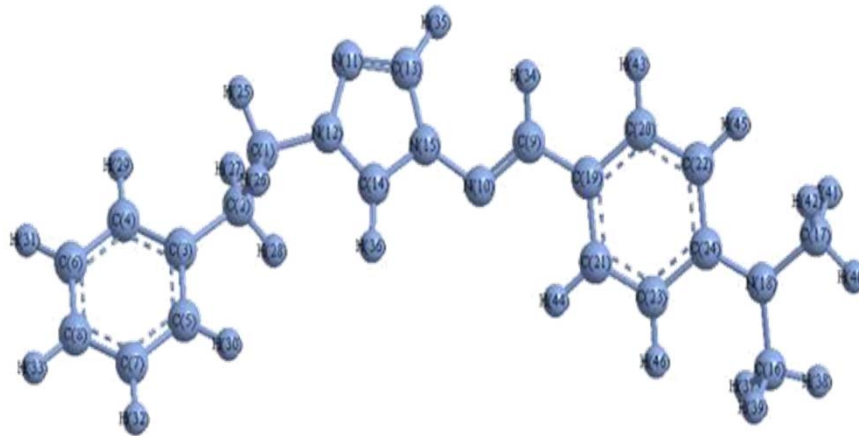
Погода у Києві [Моє місто](#)

Характеристики погоди, атмосферні явища	Температура повітря, °C	Атм. тиск, мм рт. ст.	Вітер, м/с	Вологість повітря, %	Віщується, °C
0:00 Ясно	+18	750	2 Пн	81	+18
3:00 Ясно	+17	750	2 Пд	77	+17
6:00 Ясно	+16	749	2 Пд	86	+16
9:00 Ясно	+22	750	1 Пд	59	+20
12:00 Ясно	+28	750	3 Пд	32	+26
15:00 Ясно	+29	749	3 Пд	28	+28
18:00 Ясно	+27	749	1 Пд	38	+26
21:00 Ясно	+22	749	2 Пн	47	+21

2. Необхідність великого об'єму оперативної пам'яті:

- багатовимірні задачі гідро- і газодинаміки з розрахунку течій з врахуванням різних фізичних і хімічних процесів. Розрахунок одного напрямку вимагає оперативної пам'яті **понад 10 Гбайт**),
- задачі квантової хімії.

Неемпіричні (*Ab initio*) розрахунки електронної структури молекул вимагають обчислювальних затрат, пропорційних N^4 - N^5 , де N - кількість молекул.



Молекула кардіотрилу, 46 атомів, $46^4 = 4\,477\,456$

Складність паралельних обчислень

1. Проблема «знайти» паралелелізм в послідовності вирішенні задачі.
2. Необхідність перебудови традиційної послідовної технології розв'язання задач на ЕОМ. Використовувані методи, алгоритми і системне програмне забезпечення повинні забезпечувати створення паралельних програм, організовувати синхронізацію тощо.
3. Не всі задачі можуть бути розпаралелені на конкретній архітектурі. Залежність ефективності паралелізму від характеристик паралельної обчислювальної системи на відміну від послідовної. Відтак перенесення паралельних алгоритмів і програм між різними типами систем деколи буває взагалі неможливим.

Суперкомп'ютер – обчислювальна машина, яка значно перевищує за своїми технічними параметрами більшість існуючих комп'ютерів.

Кластер – група незалежних гетерогенних обчислювальних машин із типовими апаратними та програмними рішеннями, об'єднаних в локальну обчислювальну мережу, що використовуються спільно і працюють як єдиний обчислювальний ресурс.



Суперкомп'ютер Titan,
[Oak Ridge National Laboratory](#),
(США)



Кластер [Хемніцького](#)
[технологічного університету](#)
(Німеччина)

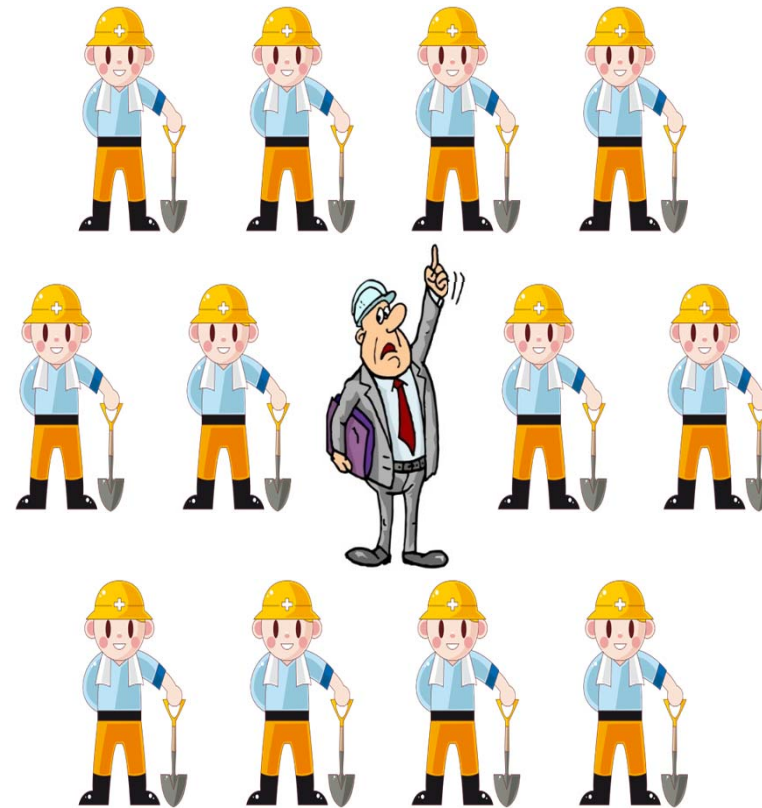


Кластер [Головної](#)
[астрономічної](#)
[обсерваторії НАН](#)
[України](#)



Паралельні обчислення на мультипроцесорній системі:

- об'єднання спеціалізованих процесорів;
- спеціальне програмне забезпечення;
- спільна пам'ять;
- надшвидке з'єднання.



Розподілені обчислення на мультикомп'ютерній системі:

- мережа універсальних (стандартних) комп'ютерів із високою швидкістю;
- розподілена пам'ять (у кожного своя);
- один або кілька керуючих вузлів.

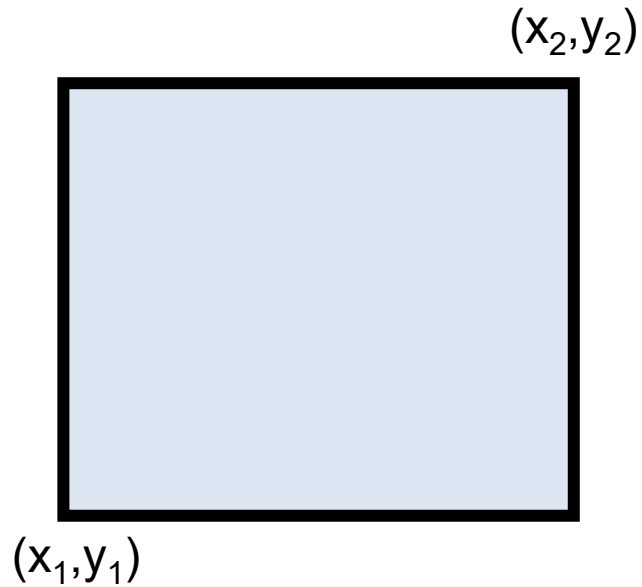
Паралельні завдання – завдання, що допускають одночасне (НЕ обов'язково незалежне) виконання.

Паралельний алгоритм – алгоритм, операції якого можуть виконуватися одночасно (не обов'язково незалежно); мається на увазі, що в явному або неявному вигляді вказані одночасно виконувані операції або набір операцій.

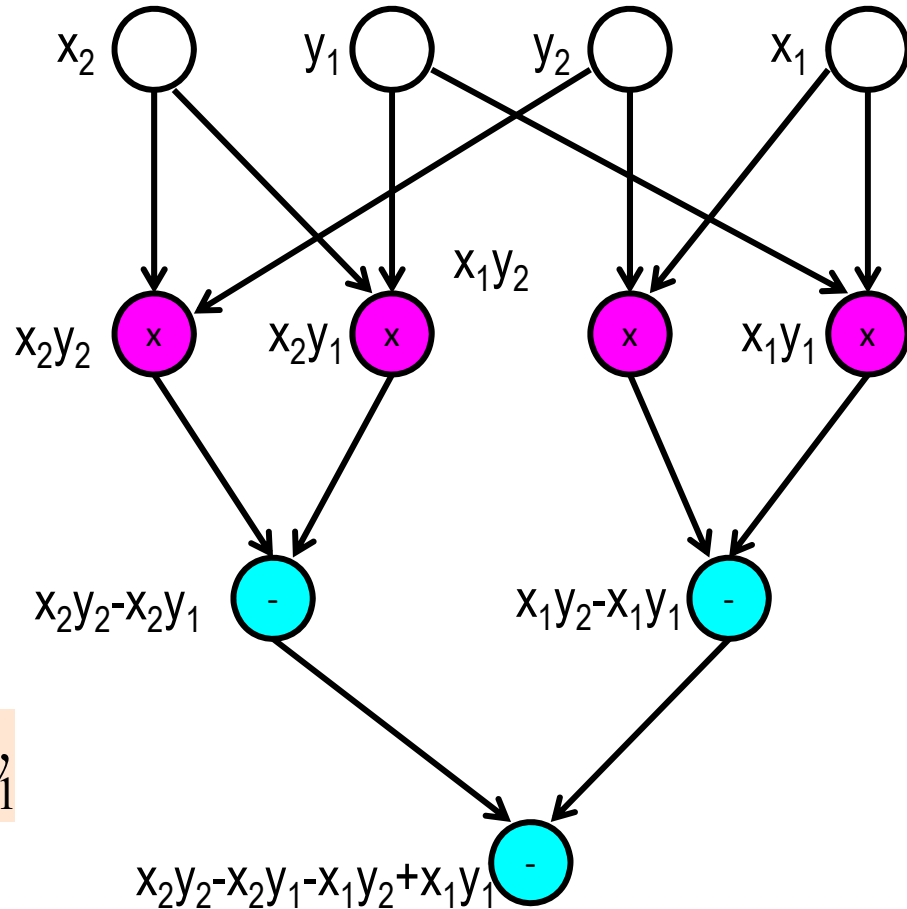
Паралельна програма – паралельний алгоритм, записаний в деякому середовищі програмування, орієнтованому на обчислювальні системи паралельної архітектури. Відповідно, техніка створення паралельних програм називається **паралельним програмуванням** (є підмножиною більш широкого поняття *багатопотоковості - multithreading*).

Паралельна система – це паралельна програма + паралельна архітектура. Паралельна програма невід'ємна від паралельної архітектури.

Приклад: обчислення площі прямокутника.



$$S = (x_2 - x_1)(y_2 - y_1) = x_2 y_2 - x_2 y_1 - x_1 y_2 + x_1 y_1$$



Приклад: сума значень у масиві

Послідовний алгоритм

```
for(i=0; i<N; i++)  
    sum+=data[i];
```

Паралельний алгоритм (нехай N - парне)

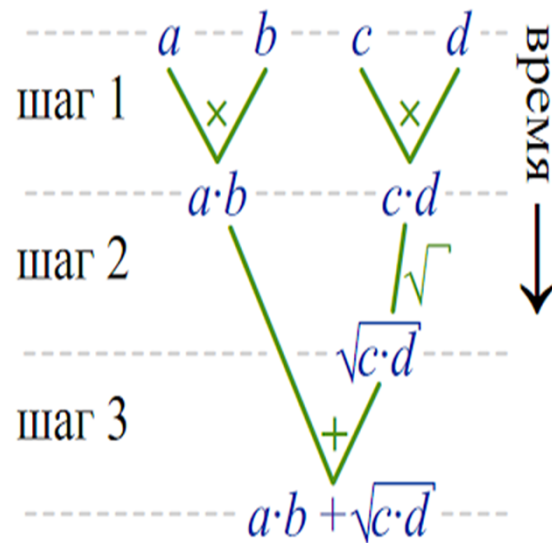
```
for(i=0; i<N/2; i++) {  
    sum1+=data[2*i];  
    sum2+=data[2*i+1];  
}  
sum=sum1+sum2;
```

Властивості паралельних алгоритмів

Розглянемо, наприклад, алгоритм обчислення виразу:

$$a \cdot b + \sqrt{c \cdot d}$$

Спочатку потрібно обчислити добуток $a \cdot b$ і $c \cdot d$, потім взяти корінь, і, нарешті, виконати додавання. Ми не можемо виконати додавання, поки не обчислені обидва його аргументу. Описаний алгоритм можна зобразити таким чином:



Видно, що процес обробки даних може бути виражений у вигляді одно направленого графа. Такий граф можна зобразити на площині, причому кожен арифметичну операцію розташовувати максимально високо (якщо вісь часу спрямована вниз), але не вище тих операцій, результат яких потрібен для її обчислення. У такому випадку висота графа буде рівна мінімальному часу (числу кроків / етапів) рішення цього завдання на ідеальній паралельній обчислювальній системі з необмеженим числом обчислювачів.

Ступінь паралелізму

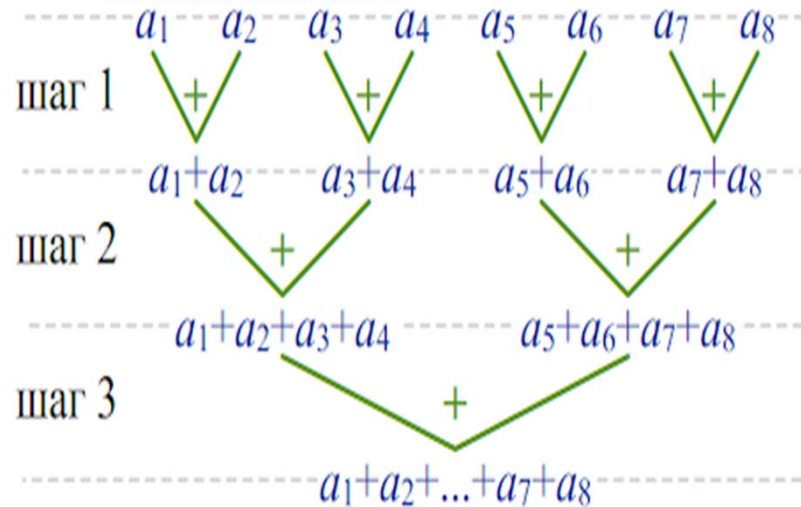
Задача додавання n чисел a_1, \dots, a_n .

Звичайний послідовний алгоритм

$$S \leftarrow a_1, \quad S \leftarrow S + a_i, \quad i = 2 \dots n$$

непридатний для паралельних обчислень. Однак завдання може бути вирішена іншим методом. На рисунку показано, як можна здійснити додавання восьми чисел в три етапи за допомогою алгоритму

здвоювання:



Завдання додавання розділено на менші підзадачі, які можуть вирішуватися незалежно. Для $n = 2^q$ чисел алгоритм здвоювання складається з $q = \log_2 n$ етапів; на першому етапі виконується $n/2$ додавань, на другому $n/4$, і так далі, поки на останньому етапі не буде виконано останнє додавання. Загальне число операцій додавання дорівнює $n - 1$: таке ж, як і в послідовному алгоритмі

Ступінь паралелізму

- Подібний алгоритм може бути застосований і для інших цілей, наприклад, для знаходження максимального елемента в масиві і навіть для сортування (сортування злиттям).
- Додавання за допомогою алгоритму здвоювання має ще одну перевагу перед послідовним додаванням: він забезпечує кращу (в середньому) точність підсумовування при використанні чисел з плаваючою крапкою.
- *Середнім ступенем паралелізму* чисельного алгоритму називається відношення загального числа операцій алгоритму до його етапів. Очевидно, для алгоритму здвоювання середня ступінь паралелізму дорівнює:

$$S = \frac{n - 1}{\log_2 n}$$

- Зі ступенем паралелізму також пов'язане поняття зернистості. Крупнозернистість завдання означає наявність в ній великих незалежних підзадач, які можна обробляти паралельно. Прикладом може служити завдання вирішення декількох різних великих систем лінійних рівнянь, рішення яких комбінуються на більш пізніх стадіях обчислювального процесу. Дрібнозернистість відповідає можливості паралельного виконання малих підзадач. Так, для складання двох векторів під задачі - це складання компонент, що мають однаковий номер.

Продуктивність паралельних обчислень

Процесорний час, необхідний для виконання програми, можна визначити за формулою:

$$T = N_i \text{CPI } t$$

де N_i — кількість машинних команд у програмі, а t — тривалість такту.

Швидкодія процесора вимірюється в MIPS (Million Instructions Per Second).

MIPS зворотно пропорційна CPI (Cycles per Instruction).

Нехай час виконання алгоритму на послідовній машині T_1 , причому T_s — час виконання послідовної частини алгоритму, а T_p — паралельної. Очевидно:

$$T_1 = T_s + T_p$$

При виконанні тієї ж програми на ідеальній паралельній машині, N незалежних гілок паралельної частини розподіляються по одній на N процесорів, тому час виконання цієї частини зменшується до величини T_p/N , а повний час виконання програми складе:

$$T_1 = T_s + T_p/N.$$

Коефіцієнт прискорення, який показує, у скільки разів швидше програма виконується на паралельній машині, ніж на послідовній, визначається формулою:

$$X = T_1/T_2 = \frac{T_s+T_p}{T_s+T_p/N} = \frac{1}{S+P/N}$$

де $S = T_s/(T_s + T_p)$ і $P = T_p/(T_s + T_p)$ — відносні частки послідовної і паралельної частин ($S + P = 1$).

Обмеження паралельних обчислень

Показано, на яке максимальне прискорення роботи програми можна розраховувати в залежності від частки послідовних обчислень та кількості доступних процесорів.

Передбачається, що паралельна секція може бути виконана без додаткових накладних витрат. Так як у програмі завжди присутня ініціалізація, введення/виведення та деякі суто послідовні дії, то недооцінювати даний фактор не можна – практично вся програма повинна виконуватись у паралельному режимі, що можна забезпечити лише після аналізу всієї програми (!)

Максимальне прискорення роботи програми в залежності від частки послідовних обчислень та кількості процесорів, що використовуються

N-proc	Доля послідовних обчислень				
	50%	25%	10%	5%	2%
2	1.33	1.60	1.82	1.90	1.96
8	1.78	2.91	4.71	5.93	7.02
32	1.94	3.66	7.80	12.55	19.75
512	1.99	3.97	9.83	19.28	45.63
2048	2.00	3.99	9.96	19.82	48.83