

Вступ до паралельних та розподілених обчислень

“Паралельні та розподілені
обчислення” Лекція 1

Задачі курсу

- Теоретичні основи роботи паралельних та розподілених систем і програм
- Технології розробки паралельних та розподілених програм
- Практичні навички роботи з паралельними та розподіленими системами
- Практичні навички розробки паралельних та розподілених програм

Для чого це потрібно?

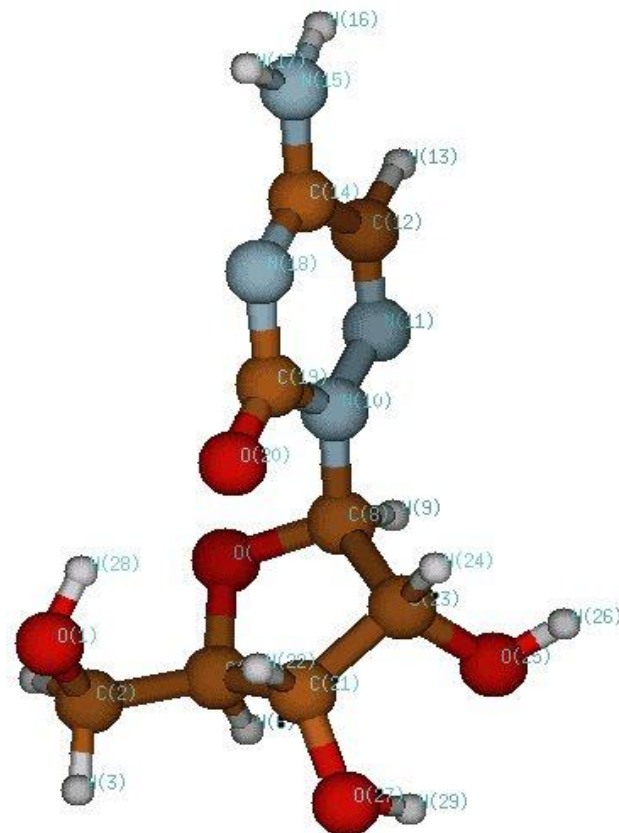
- Усі сучасні комп'ютерні системи використовують елементи паралельної обробки інформації
 - Багатопроцесорність, конвеєрна обробка ...
- Усі сучасні комп'ютерні системи використовують розподілені обчислення
 - Користувачі звикли до того, що можна працювати «одразу» із декількома комп'ютерами та програмами
 - *Інтернет, локальні мережі, зв'язані об'єкти...*
- Деякі задачі можна сьогодні вирішити лише з допомогою паралельних та розподілених обчислень
 - *отримання «надзвичайно» високої продуктивності*
 - *отримання високої надійності та відмовостійкості*
 - *деякі ресурси розподілені за визначенням*
- Фахівці повинні в цьому розбиратися

Наукові та промислові задачі, що потребують паралельних обчислень

- Хімія, молекулярна біологія
 - розробка лікарських препаратів
- Моделювання
 - метод Монте-Карло
 - Моделювання органів, систем
 - Нейросистеми
- Обробка даних
 - Набори зображень
- Без суперкомп'ютерів важко розв'язувати сучасні науково-технічні задачі

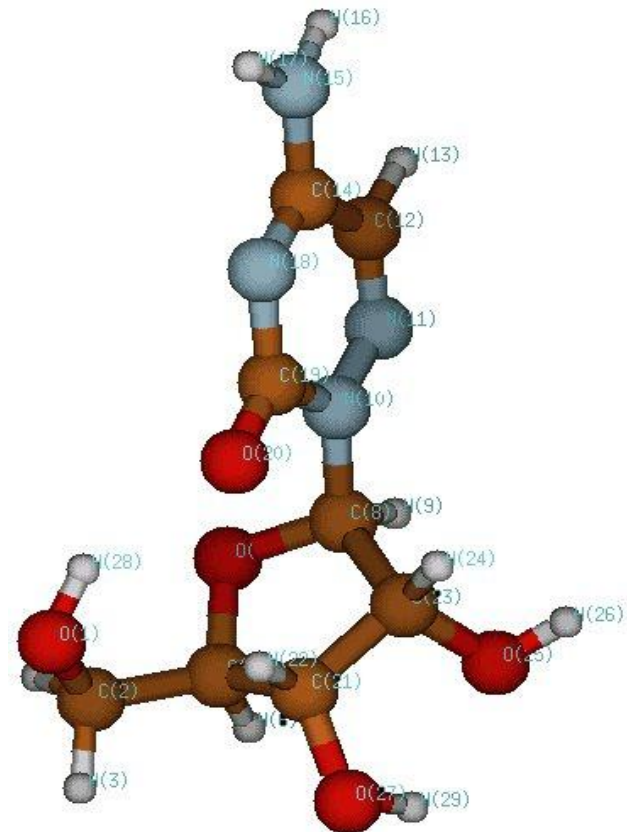
Хімія

- Є формула речовини (лікарський препарат), знайти, як ця речовина вступає в реакцію, наскільки вона стійка і як діє



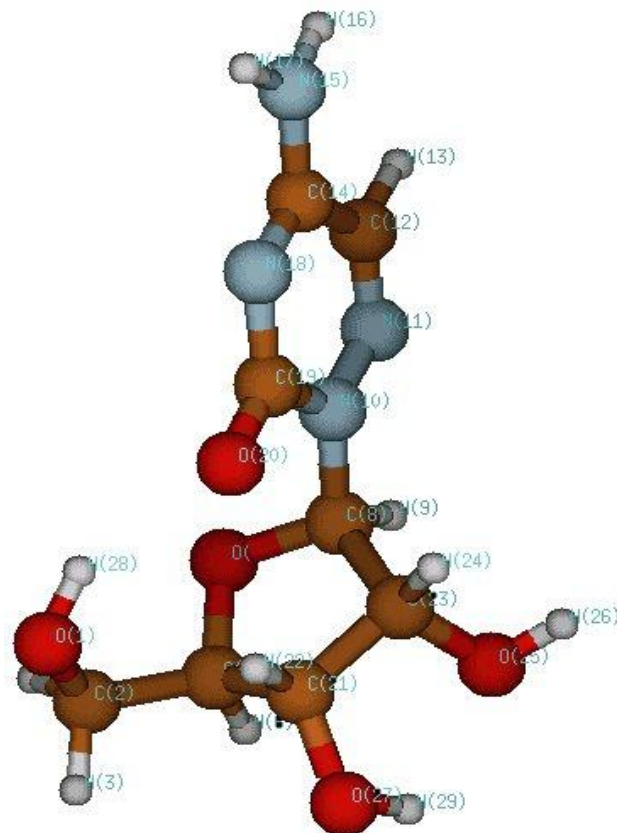
Як вирішується задача

- Властивості речовини визначаються типом атомів, положенням ядер та електронною конфігурацією
- Для знаходження електронної конфігурації необхідно розв'язувати рівняння Шредінгера для всіх атомів
- Кількість операцій, та об'єм оперативної пам'яті, необхідні для розв'язання визначаються числом електронів молекули N
- Кількість операцій пропорційна N^4 - N^7
- Об'єм оперативної пам'яті пропорційний N^3 - N^4



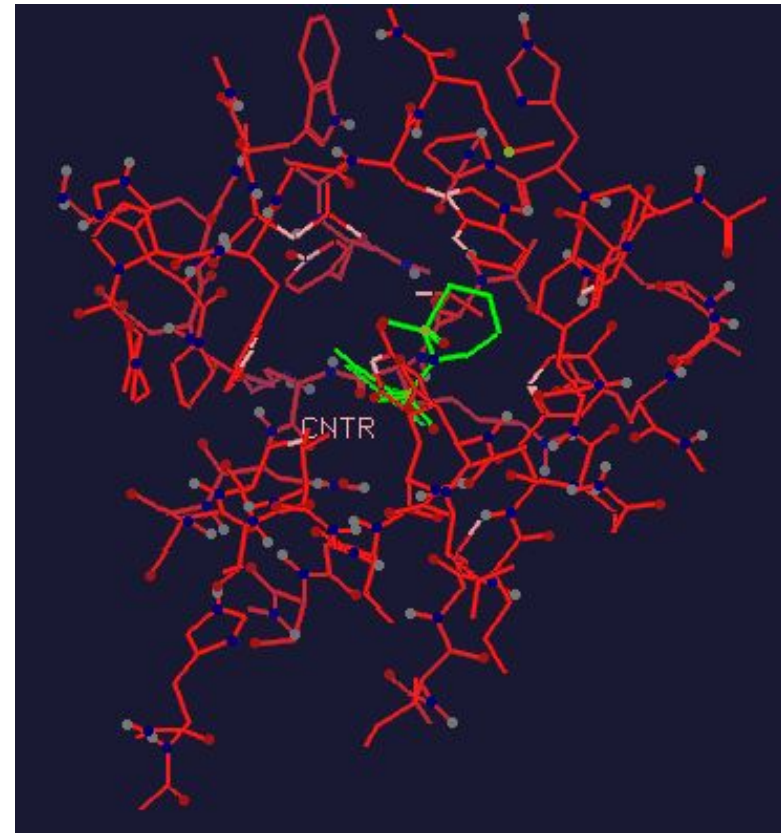
Оцінка часу та ресурсів

- Кількість атомів 29
- Кількість електронів $N=130$
- Кількість базисних функцій 280
- Кількість операцій $N^5 \sim 10^{13}$
- Задачу необхідно розв'язувати десятки/сотні разів $\sim 10^{15}$
- Час на процесорі потужністю 1 млрд. операцій за секунду
близько тижня
- Пам'яті близько 4 Гбайт
- Необхідно декілька процесорів



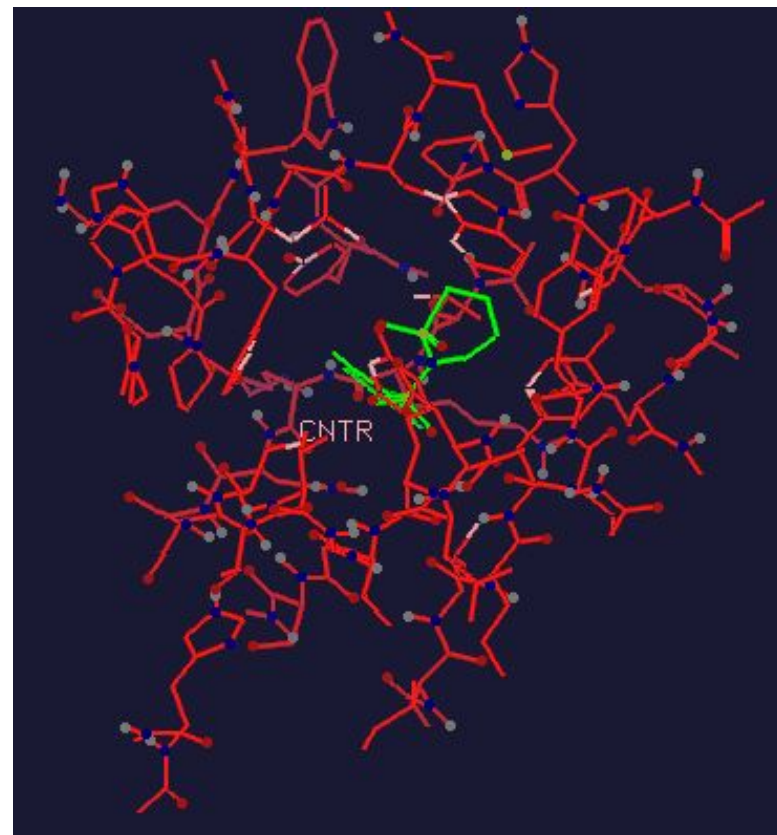
Молекулярна біохімія

- Є вірусний білок для якого потрібно підібрати лікарський препарат, який буде на нього діяти
- Кількість атомів декілька тисяч



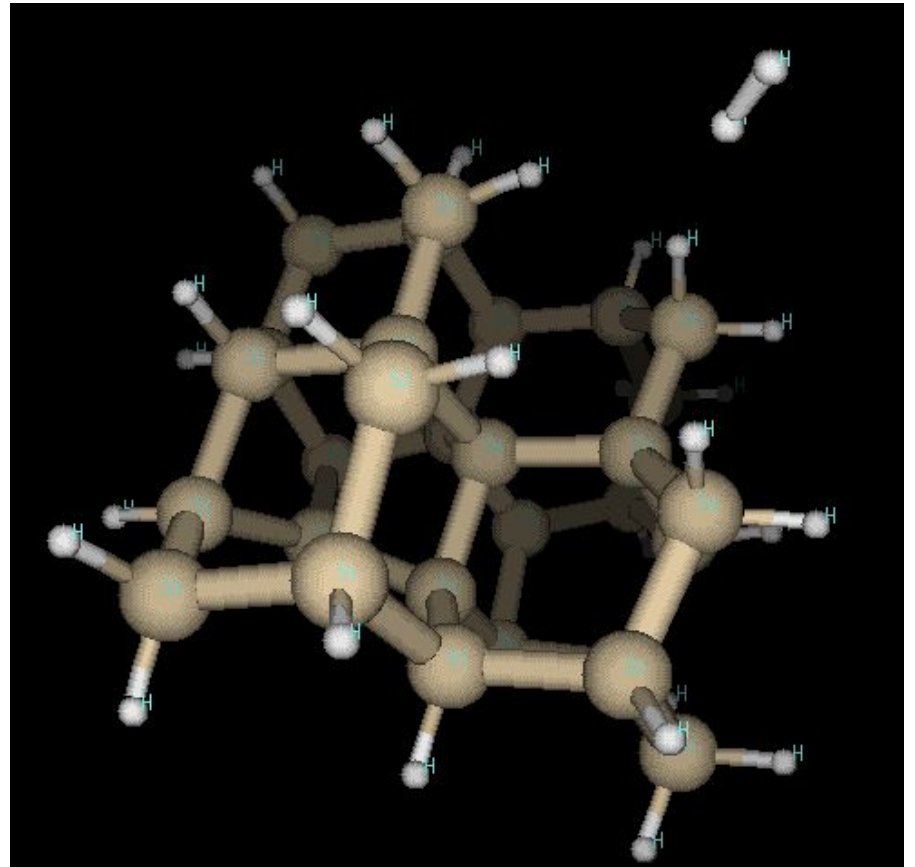
Як вирішується задача

- Використовуються наближені методи класичної фізики
- Кількість операцій MN^2 , де M кількість ітерацій, N кількість атомів
- Потребує інтенсивного обміну між процесорами
- Час розрахунку – декілька тижнів



Мікро (нано) електроніка

- Дослідження поведінки атомів на поверхні кремнію для створення нових технологій
- Потребує квантово-фізичних розрахунків



Ядерна фізика

- Взаємодія іонізуючого випромінювання з речовиною
- Моделюється поведінка великої кількості частинок
- Обробка даних з прискорювачів



Використання розподілених обчислень

- Прикладні інтернет програми
- Високонадійні системи
- Паралельні розрахунки на багатьох комп'ютерах

Література

- Параллельные вычисления в России <http://www.parallel.ru>
- Обчислювальний кластер Київського національного університету імені Тараса Шевченка <http://www.cluster.kiev.ua>
- В.П. Гергель, Р.Г. Стронгин Основы параллельных вычислений для многопроцессорных вычислительных машин. Нижний новгород: Изд-во ННГУ им. Лобачевского, 2000, 176 с.
- К. Хьюз, Т. Хьюз. Параллельное и распределенное программирование с использование C++. Перс. с англ. М: Издательский дом «Вильямс», 2004, 672 с.
- И. Н Молчанов. Введение в алгоритмы параллельных вычислений. — К.: Наукова Думка, 1990. — 128 с.
- Distributed information systems <http://www.iks.inf.ethz.ch/education/ws04/eai/>
- Грід-інфраструктура для наукових та освітніх установ України <http://grid.org.ua>
- М.В. Кононов, А.В. Мисник, С.П. Радченко, О.О. Судаков Моделювання фізичних процесів Київський університет, Київ, 2006, 90с (Укр.)
- The Linux documantation project <http://tldp.org>
- Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. - Вильямс, издательский дом, Москва, 2003, 512 с. (Рос.)
- Дейтел Х. М., Дейтел П. Д., Сантри С.И. Технологии программирования на Java 2: Бином, Москва, 2003, Кн.2, 464с. (Рос.)
- 6Geist Al, Beguelin A., Dongarra J., Jiang W., Manchek R. PVM: parallel virtual machine .- The MIT press, Cambridge; London, 1997, 278p. (Англ.)
- Eds. A.M.Bruaset, A.Tveito. Numerical solution of partial differential equations on parallel computers. - Springer, Berlin, Heidelberg, 2006, 482 p. (Англ.)

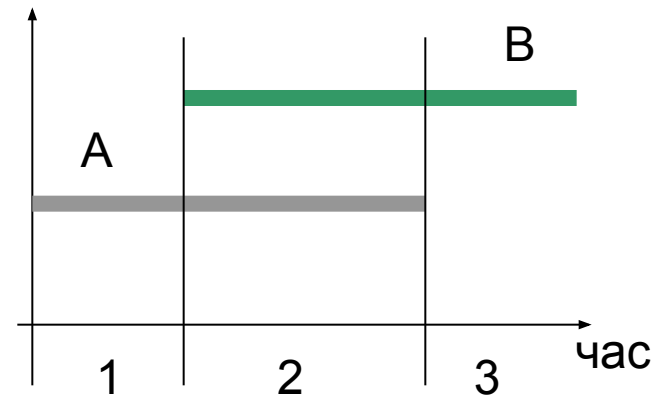
Що таке паралельні та розподілені обчислення?

Визначення паралельних та розподілених обчислень

- *Паралельні обчислення* – для розрахунку **одночасно** використовується **декілька фізичних пристроїв**
- *Розподілені обчислення* – розрахунок виконується у **декількох адресних просторах** (за допомогою **декількох процесів-програм**)

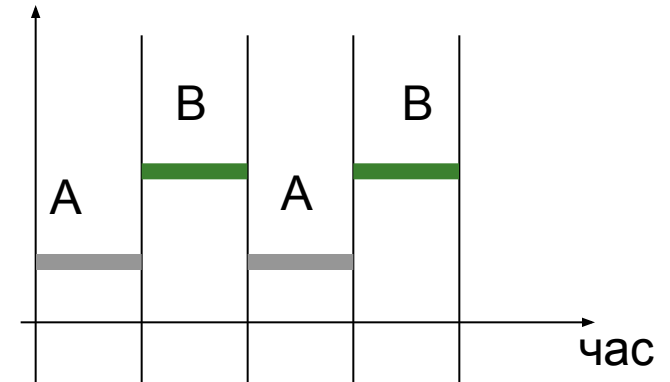
Паралельно – значить одночасно

- На проміжках часу 1 та 2 «проходить» процес А
- На проміжках часу 2 та 3 «проходить» процес В
- На проміжку часу 2 процеси А та В «проходять» одночасно, тобто паралельно



Не одночасно — значить не паралельно

- Коли процес А виконується, процес В — не виконується
- Процеси А та В виконуються не паралельно



Приклади паралельного виконання

- Двухпроцесорний комп'ютер
- Дисковий масив з декількох дисків
- Заводський конвеєр
- Бригада працівників, які копають яму
- Одночасна робота диска та процесора (асинхронний режим)
- Паралельна робота декількох відеоадаптерів

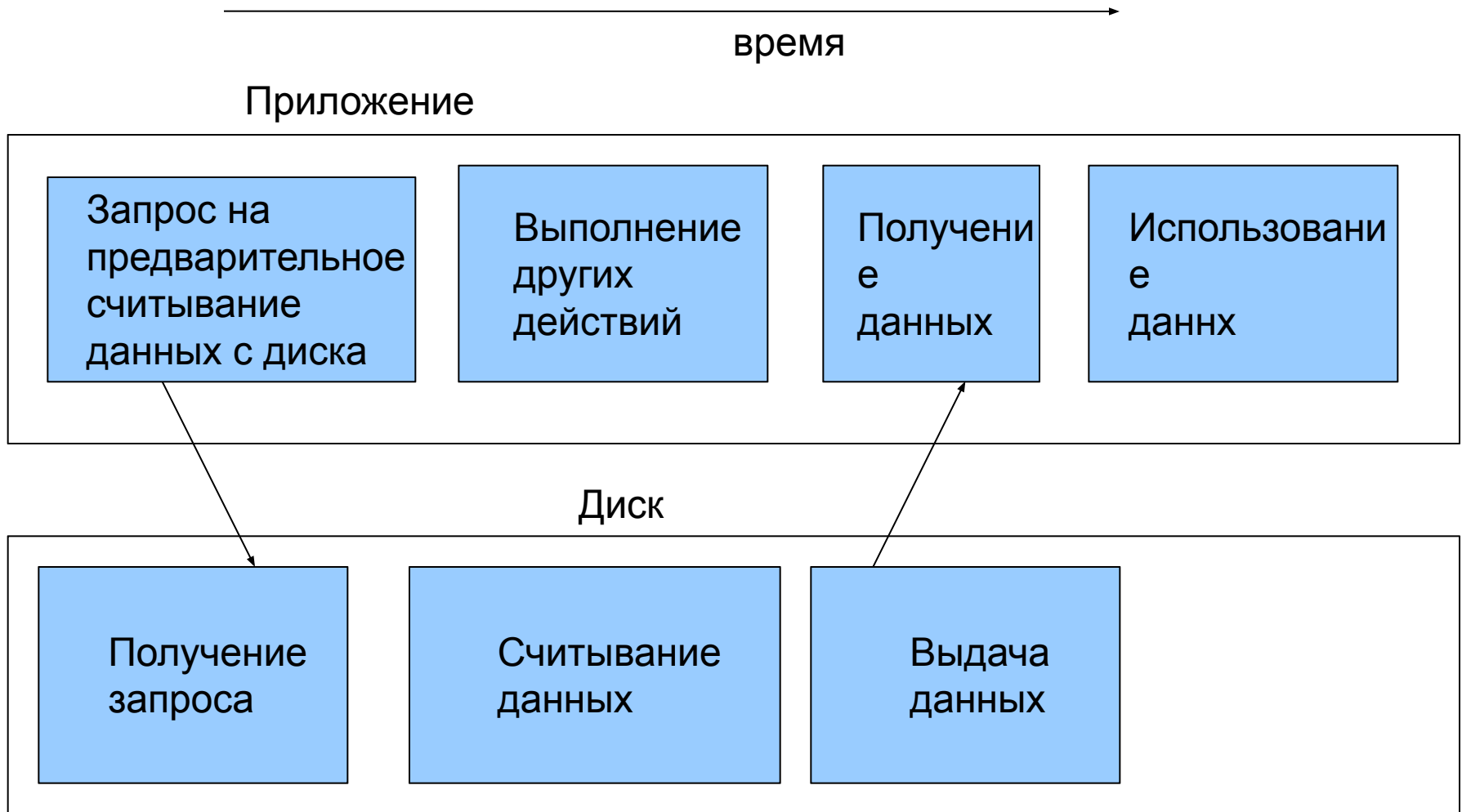
Двопроцесорний комп'ютер

```
18:49:49 up 90 days, 23:22, 6 users, load average: 1,97, 2,07, 4,93
167 processes: 161 sleeping, 3 running, 1 zombie, 2 stopped
CPU states:  cpu      user      nice    system    irq    softirq  iowait    idle
              total    197,0%    0,0%      2,2%    0,0%      0,0%      0,0%      0,4%
      cpu00      98,2%    0,0%      1,3%    0,0%      0,0%      0,0%      0,3%
      cpu01      98,8%    0,0%      0,9%    0,0%      0,0%      0,0%      0,1%
Mem    514904k av, 473604k used, 41300k free, 0k shrd, 97084k buf f
       211644k active, 202568k inactive
Swap: 2104504k av, 178928k used, 1925576k free 232668k cached

  PID USER      PRI  NI  SIZE  RSS SHARE STAT  %CPU %MEM    TIME CPU COMMAND
  654 smm        18   0 10828 10M  1572 R    99,9  2,0 561:28   1 gamess.91.x
  655 smm        14   0 10828 10M  1572 R    98,0  2,0 559:29   0 gamess.91.x
```

Кожен процесор виконує свою програму

Асинхронный режим чтения диска

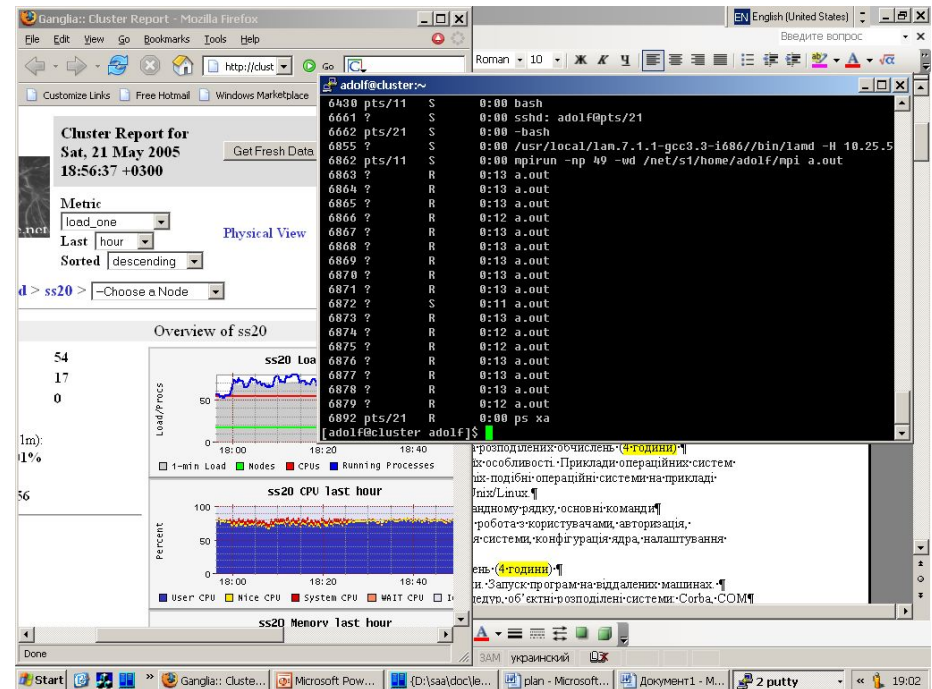


Приклади не паралельного виконання

- Багатозадачна операційна система із розділенням часу
- Мережа Ethernet із загальним середовищем передачі даних (СМАСD)
- Синхронний режим доступу до жорсткого диску

Операційна система із розділенням часу

- Кожна програма отримує свій квант часу
- Переключення між програмами виконується швидко
- Здається, що усі програми виконуються одночасно
- Насправді – не паралельно



Використання паралелізма

- Єдина мета – збільшення продуктивності

Продуктивність

- Продуктивність – кількість операцій, що виконуються за одиницю часу
- Чим складніша задача, тим більша продуктивність системи потрібна для її вирішення за розумний час
- Якщо збільшити кількість операцій, що виконуються одночасно, то збільшиться продуктивність системи

Шляхи підвищення продуктивності

■ Інтенсивні:

- ❑ Використання нових фізичних принципів побудови комп'ютерних систем (оптичні комп'ютери, наноелектроніка, високомолекулярна електроніка, квантові системи)

■ Екстенсивні:

- ❑ Збільшення тактової частоти пристроїв
- ❑ Використання паралельної обробки

Нові технології

- Найкращий варіант, але...
 - Фізичні основи сучасних комп'ютерних технологій були розроблені років 30 тому (фізика напівпровідників та діелектриків)
 - Нові фізичні методи стануть технологіями приблизно років через 30

Збільшення тактової частоти

- Продуктивність пропорційна тактовій частоті
 - Збільшення тактової частоти призводить до збільшення потрібної потужності та до необхідності посиленого охолодження
 - Збільшення тактової частоти призводить до зростання впливу паразитних обернених зв'язків та до необхідності введення нових технічних рішень

Паралельні обчислення

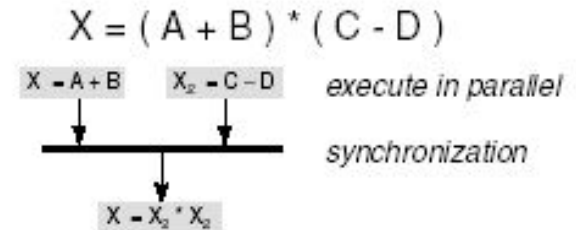
- Якщо один робітник викопав яму за 1 годину, то 2 робітники – за 30 хвилин
- Якщо один процесор повільно..., то можна поставити 2, 3, 100 ... і буде швидко
- Можна підвищувати продуктивність без введення принципово нових фізичних та технічних рішень
- Ніяким іншим методом сьогодні неможна досягти такого підвищення продуктивності, як за рахунок паралельної обробки

Рівні паралелізму

- Рівень дрібних структурних одиниць (fine graine)
 - рівень інструкцій
- На рівні середніх структурних одиниць
 - рівень підпрограм
- На рівні великих структурних одиниць (course graine)
 - Рівень об'єктів
 - Рівень прикладних програм

Паралелізм на рівні машинних інструкцій

- Дві (чи більше) машинних інструкцій виконуються одночасно
- Суперскалярні та векторні процесори
- Конвеєри
- Присутні в усіх сучасних процесорах (SSE, MMX)



Паралелізм на рівні процедур

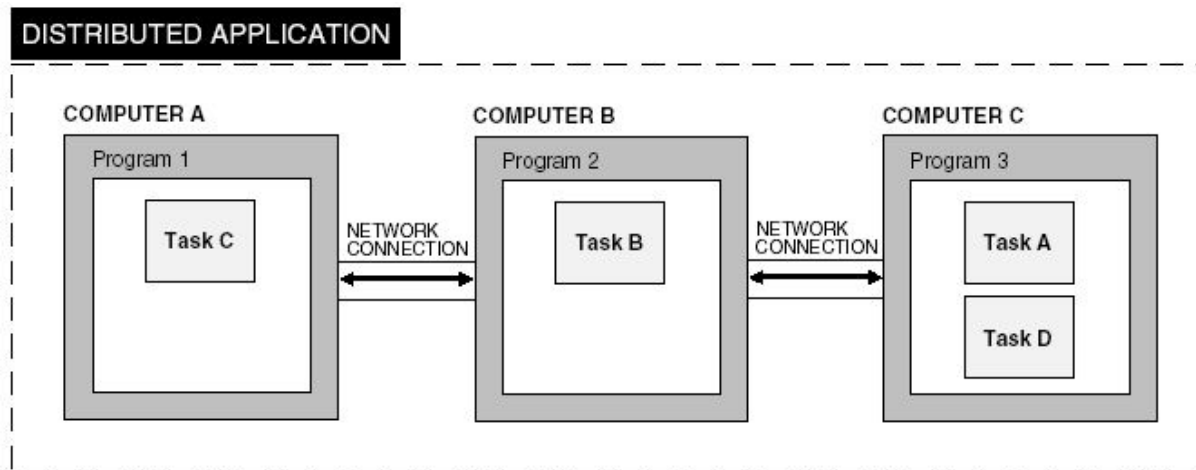
- Кожна процедура (функція, метод) окремо виконується на своєму процесорі
- Використовується при багатопотоковому програмуванні
- Потік – частина процесу, яка виконується паралельно з іншими такими ж частинами

Паралелізм на рівні об'єктів

- Методи кожного об'єкту виконуються одночасно із методами інших об'єктів
- Об'єкт – це дані та ті дії (методи, функції), котрі з цими даними можна виконувати
- Використовуються в багатопотокових програмах та розподілених об'єктних системах (COM, CORBA)

Паралелізм на рівні прикладних програм

- Кожна прикладна програма виконується на своєму процесорі або на своєму комп'ютері одночасно з іншими прикладними програмами
- Використовується для кластерних розрахунків та інших розподілених систем



Який рівень кращий?

- Для кожної задачі – свій
- Часто в одній і тій же паралельній програмі використовується одразу декілька рівнів
- Підвищення продуктивності
 - Треба підвищувати рівень
 - Збільшуються затримки
- Для уменшення задержек —
 - знижувати уровень
 - Змешнується продуктивність
- Дуже висока продуктивність – великі системи
- Дуже малі затримки – мініатюрні системи

Складності, пов'язані із паралелізмом

- Необхідність спеціальних паралельних алгоритмів
- Необхідність спеціальних паралельних програм
- Необхідність спеціальних апаратних пристроїв для паралельних розрахунків

Паралельні алгоритми

- Класичне визначення: Алгоритм – послідовність операцій, яку необхідно виконати для вирішення задачі
- Паралельний алгоритм – послідовність необхідно розбивати на послідовності, що виконуються одночасно – розпаралелити
- Дуже часто задача розпаралелення надзвичайно складна
- Іноді застосовуються власні унікальні «паралельні» підходи
- Не всі алгоритми можна ефективно розпаралелити

Декомпозиція, зв'язок та синхронізація

- Кожен паралельний алгоритм має три складові:
 - Декомпозиція
 - Зв'язок
 - Синхронізація

Декомпозиція

- **Декомпозиція** – розбиття задачі на частини, що виконуються паралельно
 - **Декомпозиція даних** – дані, з якими працює програма розбиваються на менші частини і з кожною частиною виконуються свої операції
 - **Декомпозиція функцій** – послідовність дій розбивається на ділянки, які виконуються паралельно

Приклад декомпозиції

- Розрахунок прогнозу погоди для України
- Територія розбивається на більш дрібні ділянки і для кожної ділянки виконується розрахунок на своєму процесорі, паралельно з іншими



Зв'язок

- Різні процесори повинні обмінюватися між собою інформацією

- Розрахунок прогнозу погоди для України
- Між сусідніми областями повинен виконуватись обмін інформацією про стан погоди на границі областей



Синхронізація

- Забезпечення того, що всі частини які виконуються паралельно в певні моменти часу знаходяться у визначеному стані
 - Наприклад, задача розв'язана, тільки коли всі частини що виконуються паралельно закінчать свою роботу
 - Щоб дані, які зчитані із змінної вважати коректними, потрібно гарантувати, що їх в цю змінну записали

Використання спеціальних паралельних алгоритмів

- Приклад: знайти суму $S = a_1 + a_2 + \dots + a_N$
- В такому вигляді задача суттєво послідовна
- Для розпаралелення використаємо асоціативність додавання

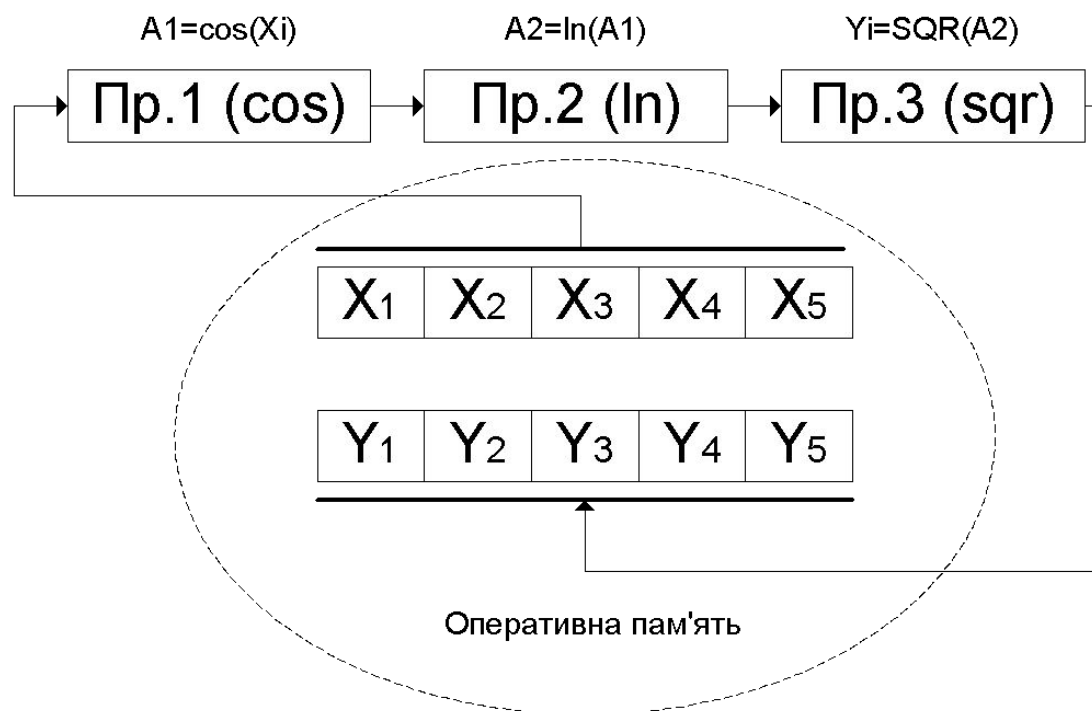
$\left. \begin{array}{l} a_1 + a_2 + \dots + a_{N/p} \\ a_{N/p+1} + a_{N/p+2} + \dots + a_{2N/p} \\ \vdots \\ a_{(p-1)N/p+1} + a_{(p-1)N/p+2} + \dots + a_N \end{array} \right\} \Rightarrow$	$\left. \begin{array}{l} S_1 \\ S_2 \\ \vdots \\ S_p \end{array} \right\} \Rightarrow$	$S = S_1 + S_2 + \dots + S_p$		S
Декомпозиція (кожен з p процесорів обраховує свою частину суми)	зв'язок (передаємо пораховані значення головному процесорові)	головний процесор додає всі результати робочих процесорів	синхронізація (чекаємо доки усі процесори не обрахували своїх суми)	видаємо результат

Приклад - конвеєр

$$y = \sqrt{\ln(\sin(x))}$$

На вході 5 значень ($X_1, X_2, \dots, X_5$)

В системі 3 процесори, кожен виконує свої дії



Результат з виходу одного процесора передаємо на вхід наступного

Стани конвеєра

На трьох процесорах результат отримується за 8 кроків замість 15 на одному процесорі

	Процесор 1 sin	Процесор 2 ln	Процесор 3 sqr	Y1	Y2	Y3	Y4	Y5
1	sin(X1)	-	-	-	-	-	-	-
2	sin(X2)	ln(sin(X1))	-	-	-	-	-	-
3	sin(X3)	ln(sin(X2))	sqr(ln(sin(X1)))	Y1	-	-	-	-
4	sin(X4)	ln(sin(X3))	sqr(ln(sin(X2)))	Y2	Y1	-	-	-
5	sin(X5)	ln(sin(X4))	sqr(ln(sin(X3)))	Y3	Y2	Y1	-	-
6	sin(X5)	ln(sin(X4))	sqr(ln(sin(X3)))	Y3	Y2	Y1	-	-
7	-	ln(sin(X5))	sqr(ln(sin(X4)))	Y4	Y3	Y2	Y1	-
8	-	-	sqr(ln(sin(X5)))	Y5	Y4	Y3	Y2	Y1

Складність

- Паралельний алгоритм значно складніший ніж послідовний
- При невеликій кількості доданків або при великій кількості процесорів можна отримати не виграш, а програш у швидкості
- При дуже великій кількості доданків і не дуже великій кількості процесорів виграш у швидкості буде істотнішим порівняно з послідовним випадком
- Ефективність розпаралелення залежить від задачі

Спеціальні паралельні програми

- Послідовна програма виконується на одному процесорі, тому не отримує ніякої переваги від паралельного виконання
- Для паралельних програм окрім самих обчислень необхідно реалізувати зв'язок та синхронізацію
- Необхідно реалізувати декомпозицію
- Для спрощення існують спеціальні компілятори та бібліотеки
- Складність відлагодження та профілювання

Апаратні засоби паралельних обчислень

- Для паралельних обчислень потрібно декілька процесорів або комп'ютерів
- Декілька процесорів/комп'ютерів завжди в сумі дорожчі, ніж один процесор/комп'ютер
- Необхідне забезпечення високошвидкісних каналів зв'язку між процесорами/комп'ютерами
- Із збільшенням кількості та складності обладнання часто зменшується його надійність

Приклади паралельних систем



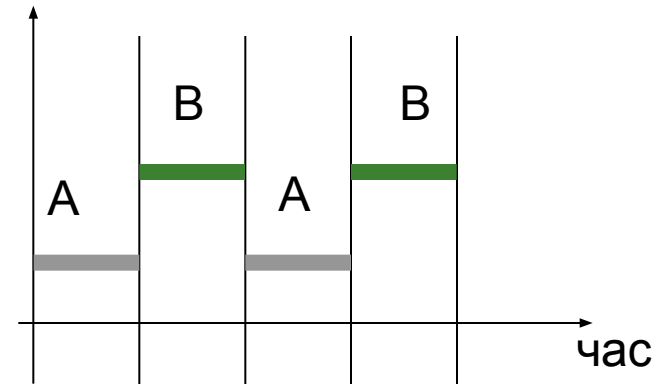
Кластер Київського національного
Університету імені Тараса Шевченка



Кластери Інституту кібернетики
НАН України імені Глушкова

Істинний та псевдопаралелізм

- Для багатозадачних операційних систем із одним процесором одночасного виконання отримати неможна, але здається, що задачі виконуються одночасно
- У такій ситуації проблеми паралелізму залишаються, а підвищення продуктивності ні - псевдопаралелізм



Паралелізм та конкуренція

- Concurrent та Parallel - синоніми
- Паралелізм – одночасність виконання задачі
- Конкуренція – одночасність використання ресурсів

Висновки відносно паралелізму

- Продуктивність послідовних ЕОМ не може зростати до нескінченності
- Єдиний спосіб отримати надзвичайно високу продуктивність на існуючому технічному рівні – це використовувати паралельні обчислення на рівні інструкцій, процедур, об'єктів, прикладних програм
- Сучасні (навіть послідовні комп'ютери) використовують паралелізм
- Використання паралельних обчислень веде до подорожчання обладнання
- Паралельні обчислення вимагають розробки спеціальних алгоритмів та використання спеціальних засобів програмування
- Не всі задачі можна ефективно розпаралелити

Розподілені обчислення

- Обчислення виконується в **декількох адресних просторах** (за допомогою **декількох процесів**)
- Процес (task) – одиниця виконання завдання, яка включає код який виконується та ресурси, які використовує цей код і які захищені від доступу інших процесів
- Адресний простір – це те, яким чином пам'ять та інші ресурси надаються процесу

Переваги розподілених систем

- Можливість використання ресурсів, які знаходяться на різних апаратних платформах або належать різним програмам
- Можливість спеціалізації ресурсів
- Можливість децентралізації
- Можливість створення надлишку ресурсів для підвищення надійності

Використання ресурсів, що знаходяться на різних комп'ютерах

- Доступ до віддалених мережевих ресурсів
- Надання доступу користувачам інших машин до своїх ресурсів
- Розподілені обчислення зараз стали синонімом слова Інтернет

Спеціалізація

- Якщо є ресурс, який необхідний великій кількості людей, то його необов'язково розміщувати на комп'ютерах усіх користувачів, яким він потрібен
- Можна створити один ресурс, до якого буде виконуватися доступ багатьох користувачів

Децентралізація

- Дані дуже великого об'єму можна рознести по декількох фізичних системах
- Приклад: великі пошукові системи

Забезпечення надійності

- Створюється декілька копій одного ресурса і на випадок виведення з ладу однієї копії, всі інші будуть доступні
- Приклад: декілька копій бази даних, або декілька задач, які виконують одні й ті ж дії

Складності

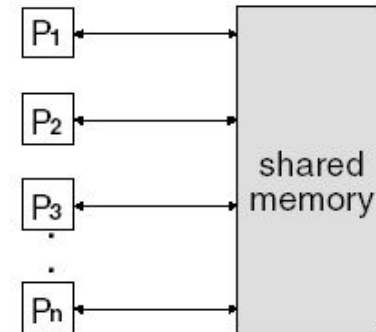
- Декомпозиція, зв'язок, синхронізація
- Ускладнення програмування

Паралельні та розподілені обчислення

- Багато спільного в цілях та підходах
- Не всі паралельні обрахунки є розподіленими
- Не всі розподілені обрахунки є паралельними

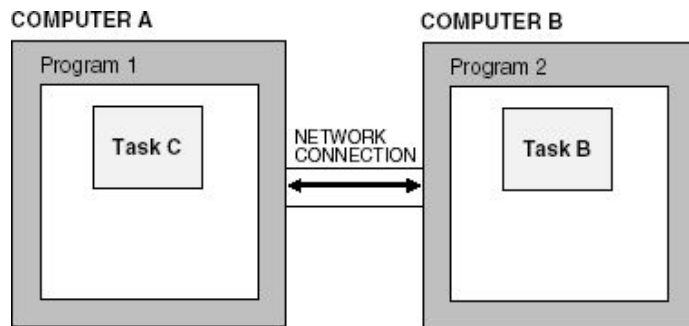
Система із спільною пам'яттю

- Паралельна, але не розподілена
- Всі програми можуть сумісно використовувати одні й ті ж самі дані



Система із дзеркалюванням

- Комп'ютер А виконує роботу, а комп'ютер В є резервним на випадок виведення комп'ютера А з ладу
- Комп'ютери не працюють одночасно



Висновки

- Паралельні та розподілені обчислення дозволяють вирішувати проблеми продуктивності, надійності та забезпечення доступу до ресурсів
- Тим не менш використання паралельних та розподілених обчислень потребує ускладнення алгоритмів, програмування та апаратних засобів
- В паралельних та розподілених обчислень багато спільного, але є деякі відмінності