



Огляд паралельних та розподілених обчислювальних систем

Паралельні та розподілені обчислення



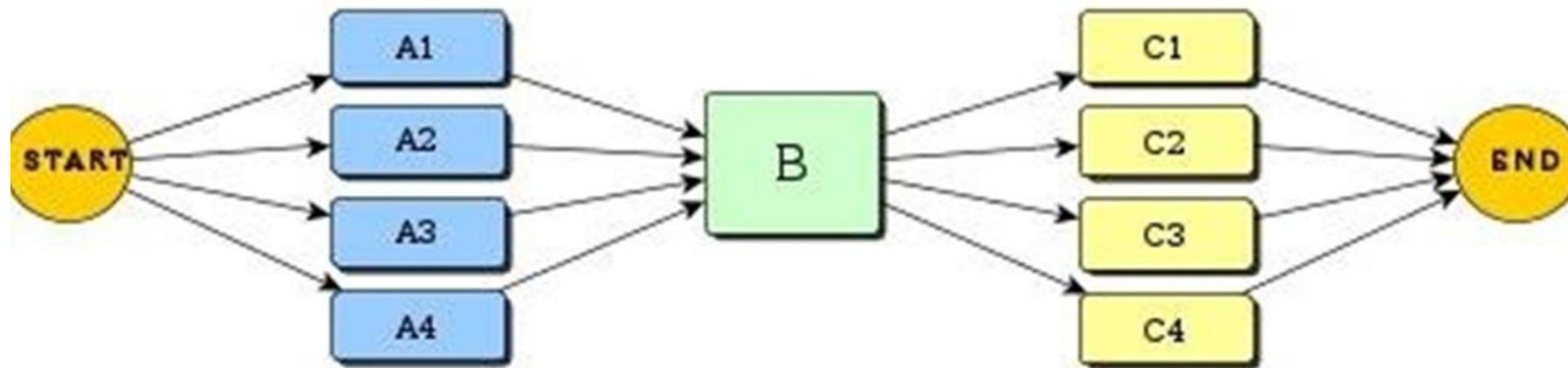
Визначення

Паралельні обчислення - метод організації обчислень, при якому програми розробляються як сукупність взаємодіючих обчислювальних процесів/потоків, що одночасно виконують різні задачі.

Розподілені обчислення - методи вирішення ресурсомістких обчислювальних задач з використанням декількох комп'ютерів, об'єднаних в обчислювальну систему

Паралельні \neq Розподілені

Паралельні обчислення

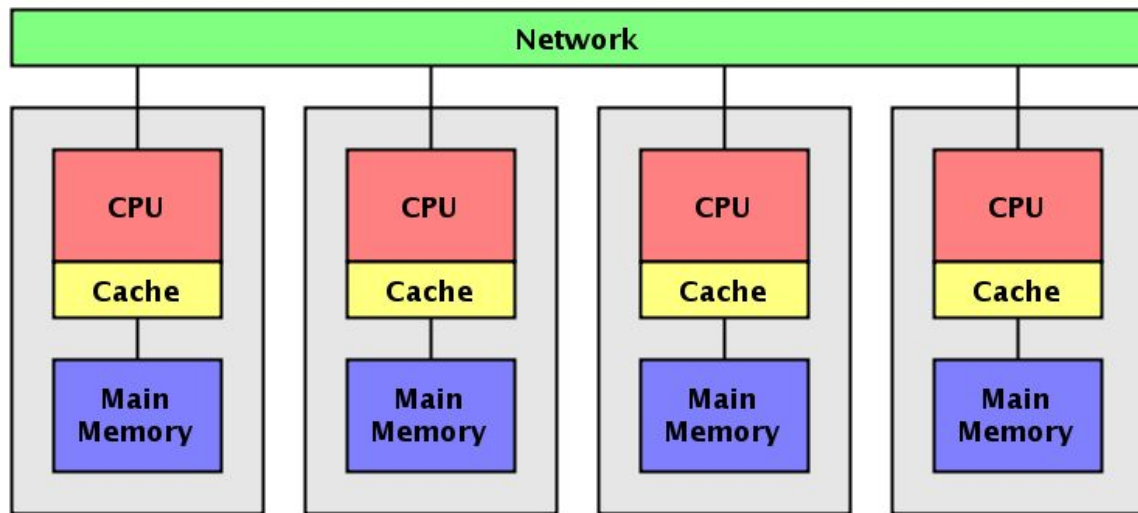


Одночасно

+ незалежно

+ кілька процесорів / комп'ютерів

Розподілені обчислення



Мережа (локальна
або глобальна)

Типи паралелізму

Паралелізм на рівні бітів (bit-level parallelism)

- 8-, 16, 32, 64, 128-розрядні процесори

Паралелізм на рівні інструкцій (Instruction-level parallelism)

- Конвейєр
- Суперскаляр
- Явний паралелізм команд (VLIW і EPIC)

Паралелізм завдань (task parallelism)

Паралелізм даних (data parallelism)

Класифікація Фліна (1970)

SISD

single instruction + single datastream - одна команда + один потік даних

SIMD

single instruction + multiple datastreams - одна команда + багато потоків даних

MISD

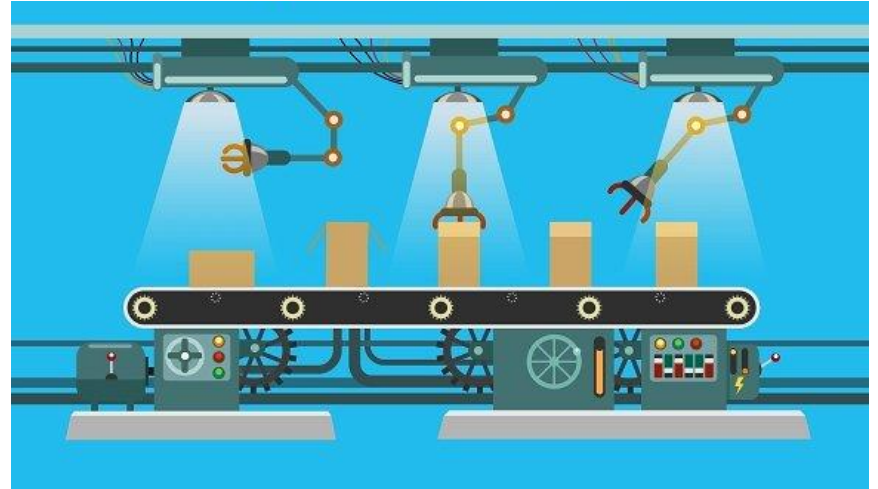
multiple instructions + single datastream - багато команд + один потік даних

MIMD

multiple instructions + multiple datastreams - багато команд + багато потоків даних

Конвейер

Обробку машинної команди розбити на незалежні етапи, які суміщати в часі для декількох команд в відповідній апаратурі (конвеєрі команд)



Конвейєр

Вибірка (instruction fetch, IF) —
завантаження нової команди з пам'яті



Декодування (instruction decode, ID) — інтерпретація та відправка команди у відповідний операційний пристрій в залежності від операції

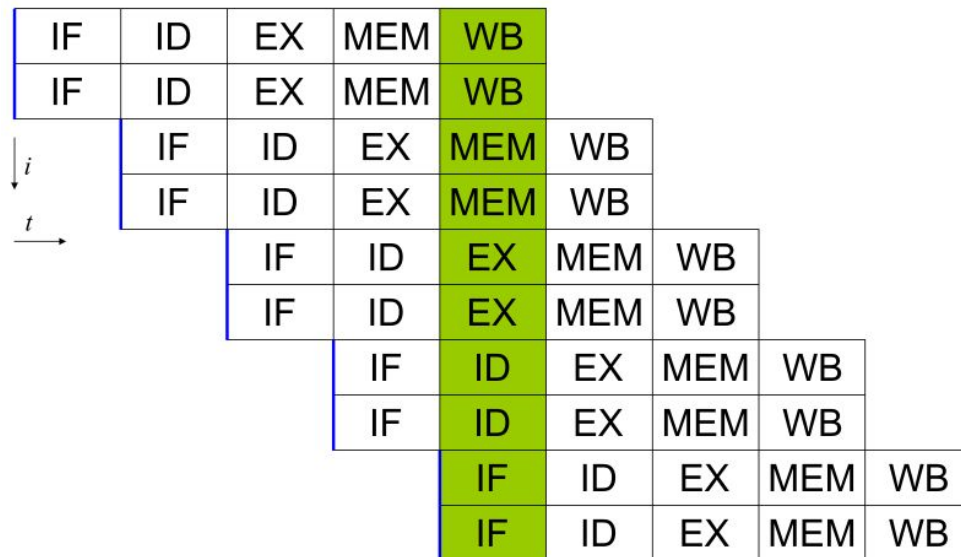
Виконання (execution, EX) — виконання команд та обчислення ефективної адреси пам'яті для результату або операндів, які необхідно завантажити

Звертання до пам'яті (memory, MEM) — операції з пам'яттю (для команд завантаження/збереження)

Збереження результату (writeback, WB) — збереження результату обчислень в регістрі

Суперскалярність

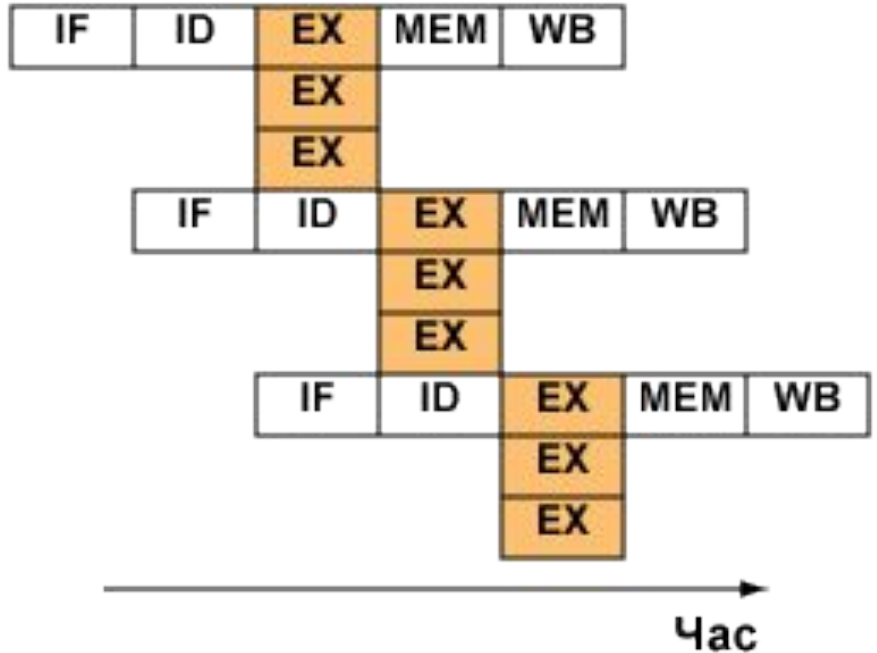
Кілька декодерів команд, навантажують декілька виконавчих блоків і т.д. Планування виконання потоку команд є динамічним і здійснюється самим обчислювальним ядром.



VLIW-конвеєр

Виконання команд -
найтриваліша операція

Виконання комплексу команд,
упакованих в довге командне
слово, на стадії EX відбувається
одночасно



Explicitly parallel instruction computing (EPIC)

Перенесення складної логіки планування інструкцій з мікропроцесора в компілятор, який планує інструкції статично.

EPIC дозволяє мікропроцесору виконувати інструкції паралельно, спираючись на роботу компілятора, а не виявляючи можливість паралельної роботи інструкцій за допомогою спеціальних схем.

Паралелізм завдань

Кожен процесор виконує окрему нитку (потік). Нитки (потоки) можуть виконувати один і той же або інший код.

Потоки взаємодіють під час виконання.

Розпаралелювання на цьому рівні є найпростішим і найефективнішим. Таке розпаралелювання можливе в тих випадках, коли розв'язуване завдання складається з незалежних підзавдань, кожне з яких можна виконати окремо.

Операційна система : різні програми на різних ядрах.

Паралелізм даних

Кожен процесор виконує одне завдання на різних частинах розподілених даних

```
for i from lower_limit to upper_limit  
    foo(d[i])
```

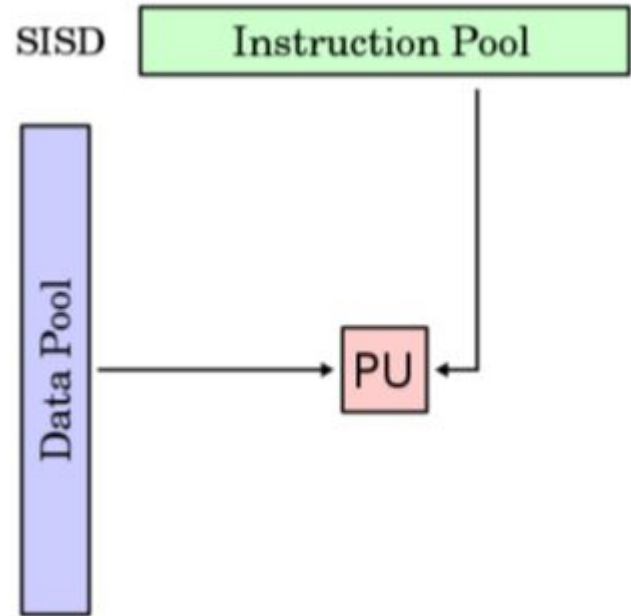
SISD

Single instruction + single datastream -
одна команда + один потік даних

Один обчислювач виконує один потік команд, оперуючи одним потоком даних.

Паралелізму немає

Архітектура фон Неймана



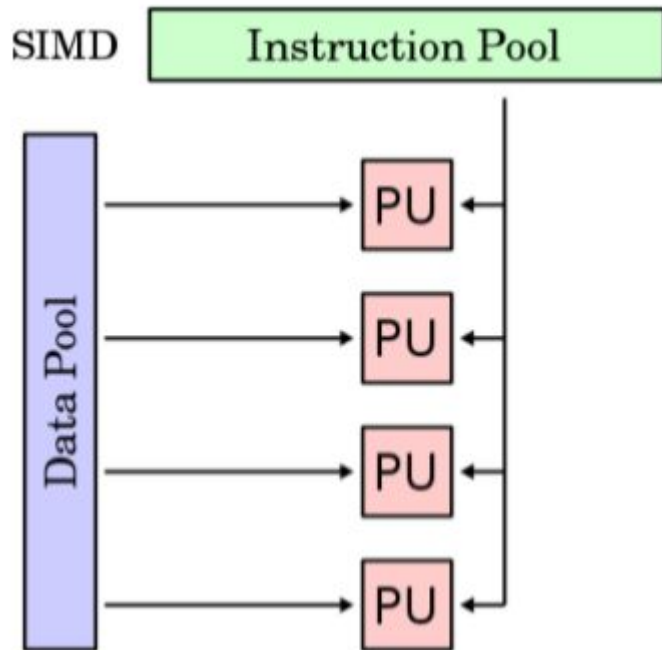
SIMD

Single instruction + multiple datastreams -одна команда + багато потоків даних

Кілька обчислювачів виконує однакові інструкції над різними потоками даних

Приклади:

- Графічні процесори GPU: архітектура CUDA і AMD FireStream
- Векторні обчислювачі



MISD

Multiple instructions + single datastream - багато команд + один потік даних

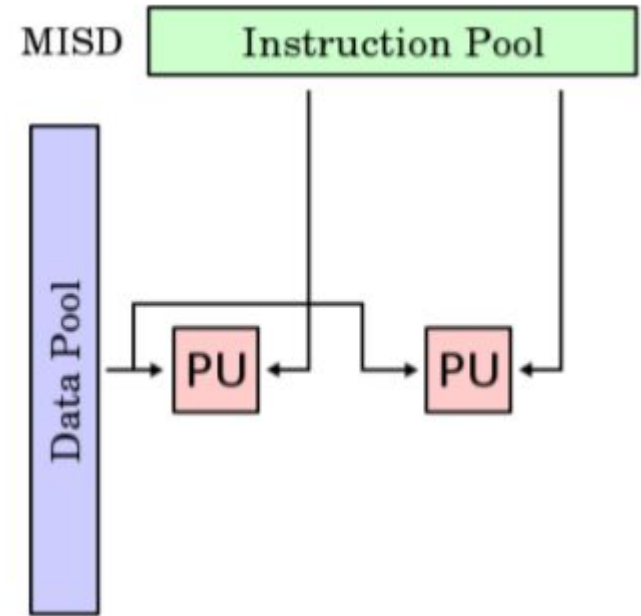
Кілька обчислювачів виконують різні інструкції над одним потоком даних

У високонадійних системах:

- Структурне резервування
- Алгоритмічне резервування

Систолічні архітектури

ВС типу MISD використовуються рідко

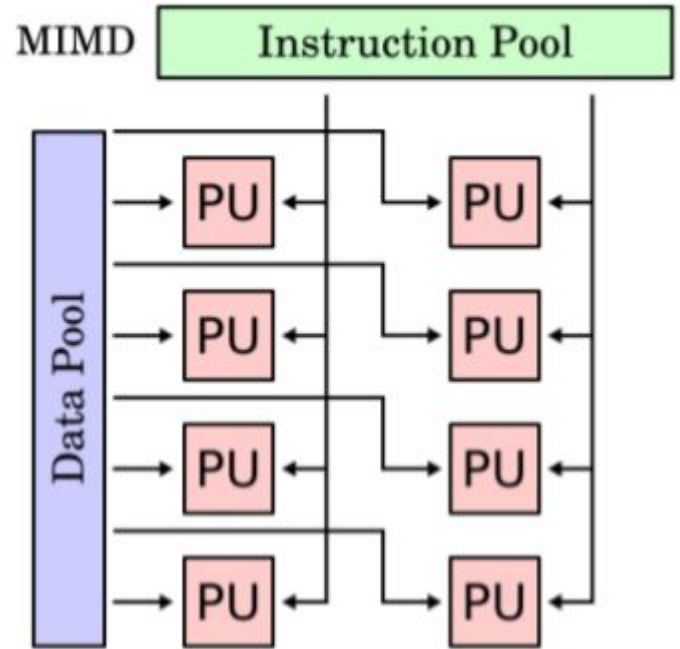


MIMD

Multiple instructions + multiple datastreams - багато команд + багато потоків даних

Кілька обчислювачів виконує різні інструкції над різними потоками даних

Найбільш загальний варіант паралелізму



Symmetric Multiprocessing (SMP)

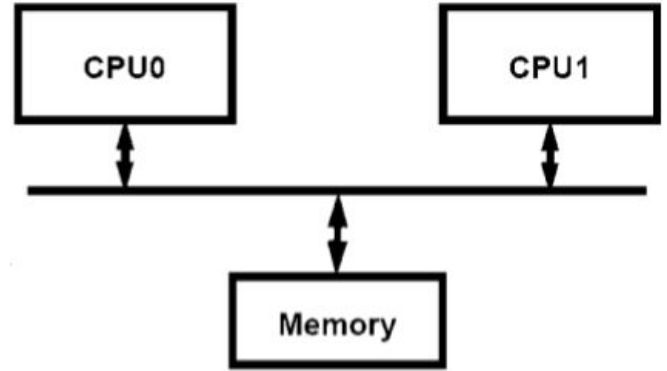
Кілька однакових процесорів підключаються до загальної пам'яті

переваги:

- Можна використовувати програми для однопроцесорних ЕОМ

проблеми:

- Обмеження на число процесорів
- Когерентність кеш-пам'яті



Massive Parallel Processing (MPP)

Архітектура з розподіленою пам'яттю

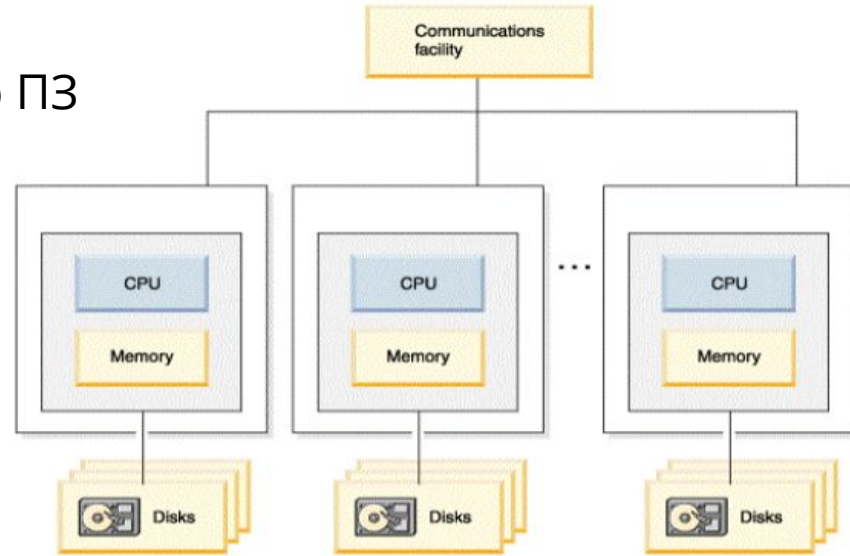
Кілька варіантів архітектури системного ПЗ

Переваги:

- Немає необхідності потактової синхронізації процесорів
- Хороша масштабованість

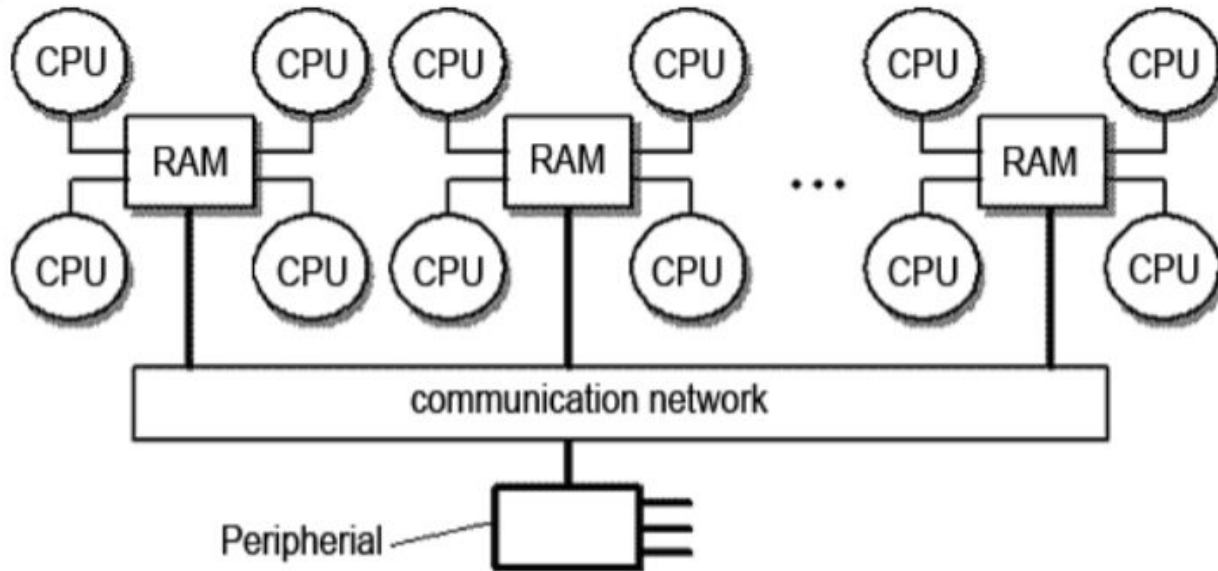
Недоліки:

- Обмежений обсяг пам'яті на процесор
- Складнощі міжпроцесорного обміну
- Трудомістка розробка ПЗ



Non-Uniform Memory Architecture (NUMA)

Складна архітектура ОС, що забезпечує розподіл завдань і даних між процесорними модулями



Гіперпоточність (hyper-threading)

Виконання 2-х потоків
на одному процесорному ядрі

Фізичний процесор складається з двох логічних

- Intel Pentium 4
- Intel Core i3/i5/i7
- Intel Xeon

Логічний процесор має свій набір регістрів і контролер переривань

Обидва логічних процесора поділяють ресурси фізичного процесора

Другий потік виконується, коли перший очікує (промах кеша, помилки в передбаченні розгалужень і т.п.)

Дає вигоду в продуктивності від 5% до 30%

Симетричні багатоядерні процесори

Містять кілька (2-12) однакових ядер

Кожне ядро має свій кеш 1-го рівня

Кеш 2-го рівня може бути спільний або індивідуальний

Призначені для запуску багатопоточних додатків

Intel Core Duo/Quad

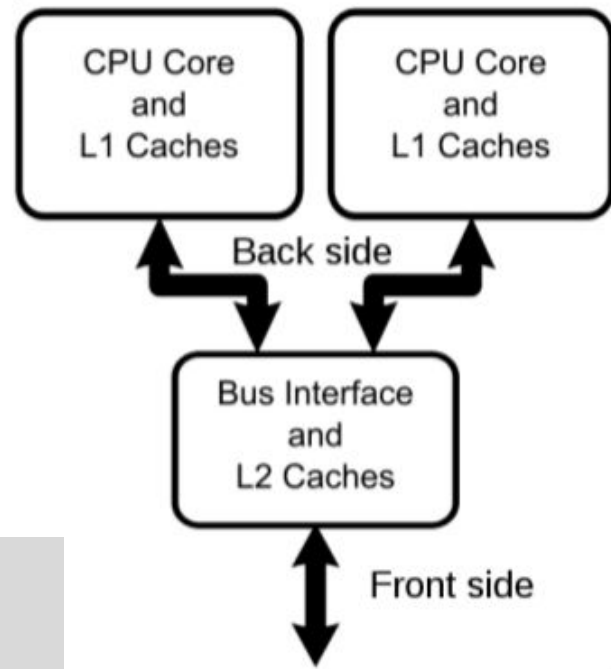
Intel Core i3/i5/i7

AMD Phenom/Phenom II

Intel Core i3/i5/i7

Intel Xeon

AMD Opteron



Процесори з асиметричними ядрами

Процесор складається з декількох ядер:

- 1 керуюче ядро
- 1 або кілька спеціалізованих обчислювальних ядер

Керуюче ядро виконує програму загального призначення

Обчислювальні ядра вирішують спеціалізовану обчислювальну задачу:

- FPU
- DSP
- GPU

Графічні процесори

Призначені для графічних обчислень, наприклад прискорення 3D-графіки.

Тенденція - GPU - швидкий співпроцесор для загальних обчислень.

- Архітектура SIMD
- GPU може виконувати паралельно тисячі ниток
- Маленька внутрішня пам'ять
- Відносно низька швидкість обміну даними з CPU
- На відповідних завданнях дає прискорення 10-100 разів порівняно з CPU

Технологія CUDA: Nvidia GeForce / Nvidia Quadro / Nvidia Tesla

Технологія ATI Stream Technology

Розподілені системи



Види розподілених систем

- P2P
- Client-Server
- Dataflow

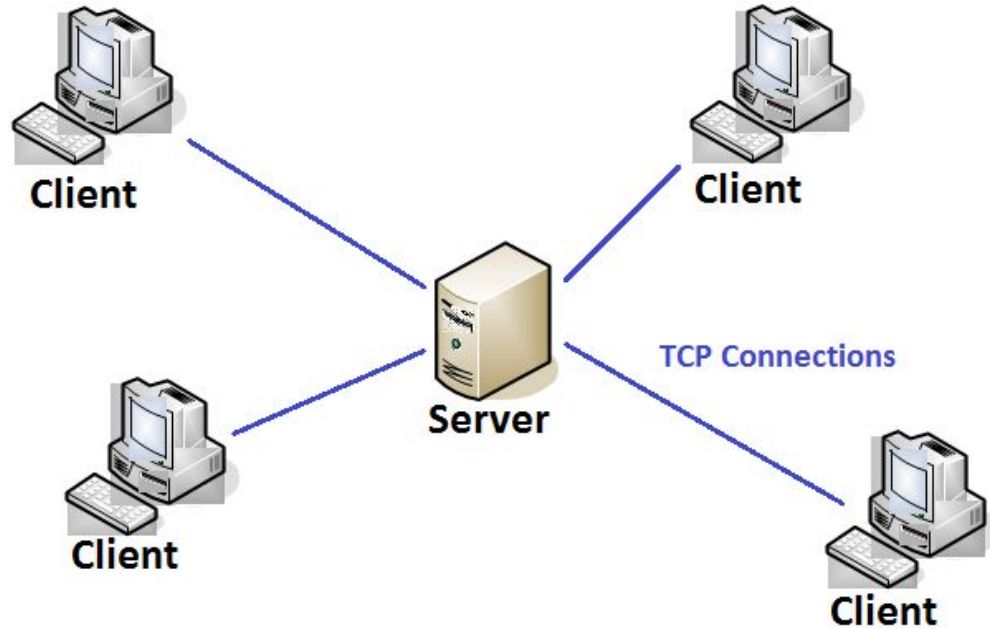
P2P (peer-to-peer)

Система рівноправних учасників

Client-Server

Клієнт ініціює комунікацію -
сервер відповідає

Протоколи: HTTP MySQL REST
SOAP



Dataflow

Учасник розподіленої системи отримує / відправляє дані

Активність

Реактивність

Stateless

