

CHƯƠNG 1: GIỚI THIỆU

YÊU CẦU VỀ TÍNH TOÁN

NHIỀU VẤN ĐỀ TRONG THỰC TIỄN

- Yêu cầu về tốc độ tính toán
- Xử lý và tính toán dữ liệu khổng lồ
 - Mô hình hóa quá trình tự nhiên; mô phỏng trong các lĩnh vực khoa học, kỹ thuật, kinh tế.
 - Dự báo thời tiết; thiên văn; sinh học; ...
 - Khai khoáng dữ liệu (Data Mining); thị giác máy tính (Computer Vision); xử lý ảnh (Image Processing), ...

YÊU CẦU VỀ TÍNH TOÁN

NHIỀU VẤN ĐỀ TRONG THỰC TIỄN

- Máy tính đơn: cần nhiều thời gian
- Máy tính song song (nhiều CPU), PC cluster (nhóm máy tính), Grid (lưới): thực hiện song song trên nhiều CPU → tăng tốc (thời gian tính toán ngắn)

YÊU CẦU VỀ TÍNH TOÁN

DỰ BÁO THỜI TIẾT

- Bài toán dự báo thời tiết: khí quyển được mô hình hóa bằng cách phân chia thành những miền hay thành những phần tử 3 chiều
- → Sử dụng các phương trình toán học để tìm ra nhiệt độ, áp suất, độ ẩm, tốc độ và hướng gió

YÊU CẦU VỀ TÍNH TOÁN

- 1 Pflop/s (PetaFlop/s) = 1.000 Tflop/s
- 1 Tflop/s (TetraFlop/s) = 1.000 Gflop/s
- 1 Gflop/s (GigaFlop/s) = 1.000 Mflop/s
- 1 Mflop/s (MegaFlop/s) = 1.000 Kflop/s
- 1 Kflop/s (KiloFlop/s) = 1.000 Flop/s
- 1 Flop/s = 1 Floating Point Operation per Second

YÊU CẦU VỀ TÍNH TOÁN

DỰ BÁO THỜI TIẾT

- Giả sử mỗi phần tử tính toán cần 200 phép tính (float point operation), với 1 vùng 5×10^8 phần tử
 - ➔ cần $200 \times 5 \times 10^8 = 10^{11}$ phép tính
- Giả sử cần dự báo thời tiết 10 ngày sau và 10 phút 1 lần tính
 - ➔ cần $(60 \times 24 \times 10)/10 = 1440$ bước thời gian
 - ➔ cần $1440 \times 10^{11} \approx 1.44 \times 10^{14}$ phép tính

YÊU CẦU VỀ TÍNH TOÁN

DỰ BÁO THỜI TIẾT

→ cần $1440 \times 10^{11} \approx 1.44 \times 10^{14}$ phép tính

➤ Với máy tính có khả năng thực hiện $100 \text{ Mflop/s} = 10^2 \text{ Mflop/s} = 10^8 \text{ Flop/s}$

➤ → cần $(1.44 \times 10^{14}) / 10^8 = 1.44 \times 10^6 \text{ second} \approx 17 \text{ ngày đêm}$ mới tính toán xong

➤ → sau 17 ngày mới biết được kết quả của ngày thứ 10

YÊU CẦU VỀ TÍNH TOÁN

DỰ BÁO THỜI TIẾT

→ cần $1440 \times 10^{11} \approx 1.44 \times 10^{14}$ phép tính

➤ Nếu yêu cầu đặt ra là phải thực hiện tính toán trong vòng 1 giờ = 3600 giây

➤ → cần $(1.44 \times 10^{14})/3600 = 0.4 \times 10^{11}$ phép tính trên giây

➤ → cần máy tính có thể thực hiện 0.4×10^{11} phép tính trên giây = 40 Gflop/s = 40 tỷ phép tính trên giây

XU HƯỚNG MÁY TÍNH

➤ Có 02 xu hướng khai thác năng lực máy tính trong việc giải quyết vấn đề khoa học tính toán:

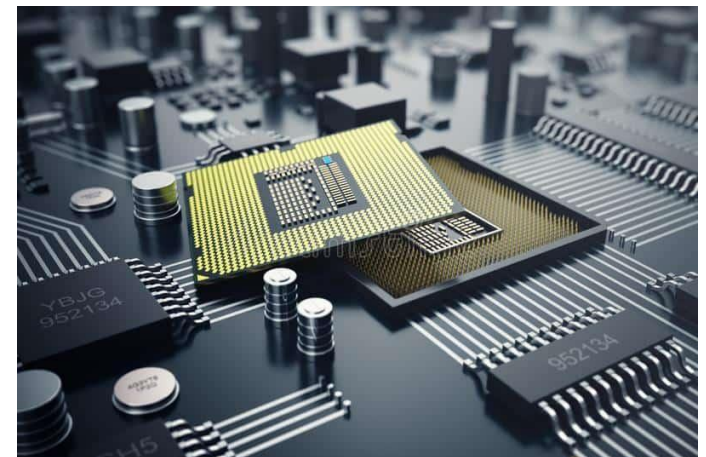
- Năng lực nội tại của một máy
- Năng lực của hệ thống máy



NĂNG LỰC NỘI TẠİ CỦA MỘT MÁY TÍNH

➤ 02 hướng phát triển chính

- Có được nhiều bộ xử lý, nhưng không tốn nhiều năng lượng hơn.
- Tận dụng những gì đã có để tăng khả năng xử lý



NĂNG LỰC NỘI TẠI CỦA MỘT MÁY TÍNH

- Máy tính vẫn mạnh nhưng
 - ❑ Nguồn năng lượng cung cấp không tăng lên
 - ❑ Tỏa nhiệt cũng không nhiều hơn



Desktop



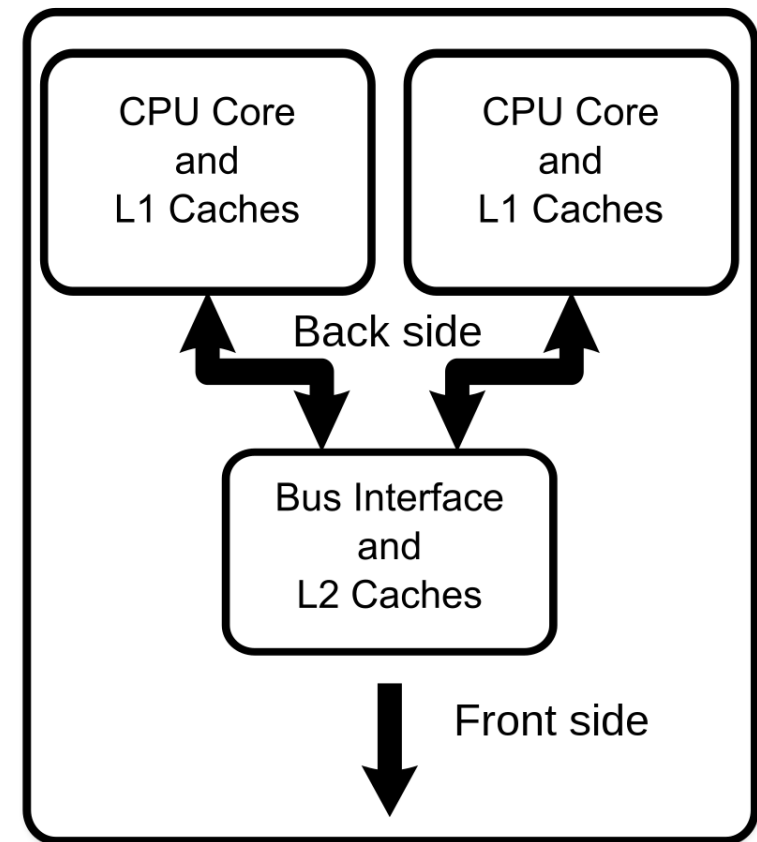
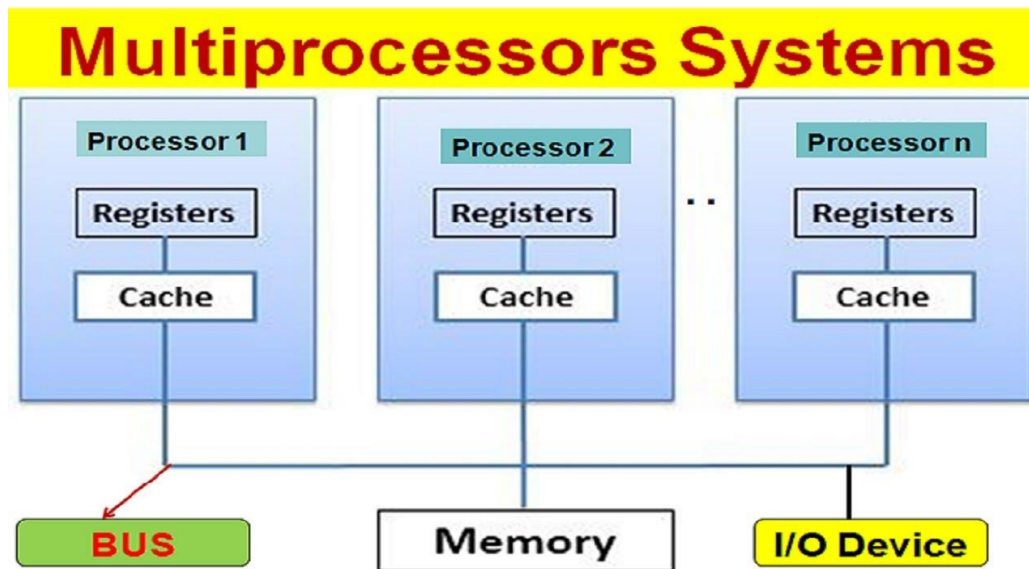
Laptop



Netbook

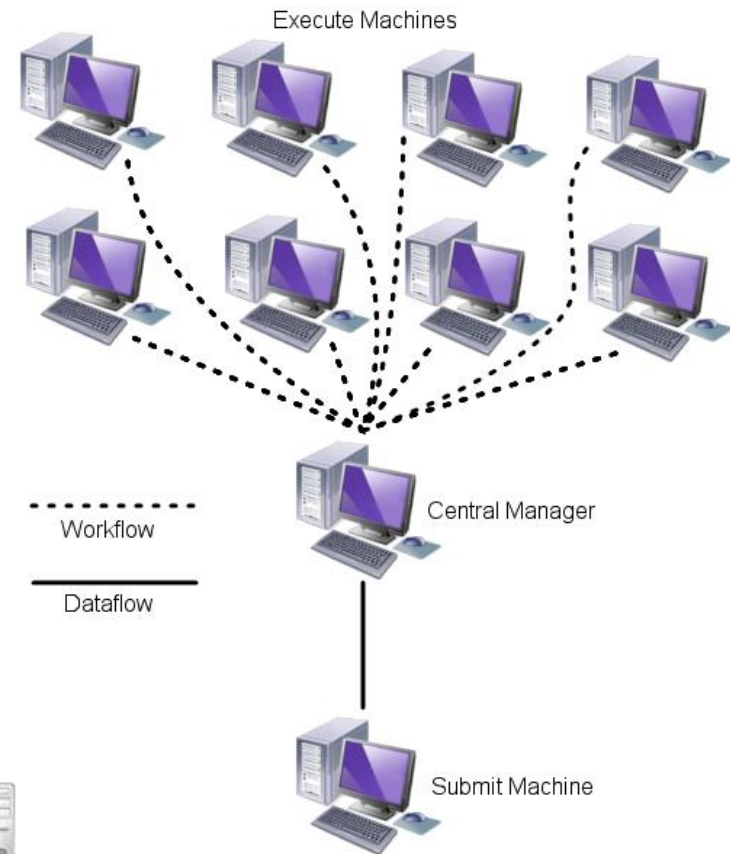
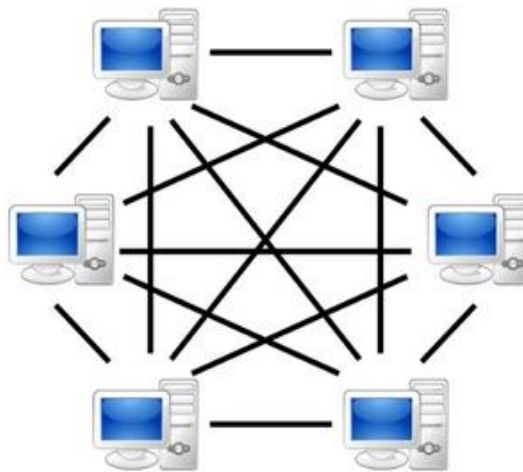
NĂNG LỰC NỘI TẠİ CỦA MỘT MÁY TÍNH

- Trên laptop hoặc netbook, thay vì Multi-Processor, chuyển sang Single-Processor nhưng với Multi-Core



NĂNG LỰC HỆ THỐNG MÁY

- Distributed Computing
- Cluster Computing
- Peer to peer Computing
- Ambiguous Computing
- Volunteer Computing
- Grid Computing
- Cloud Computing
- Sky Computing



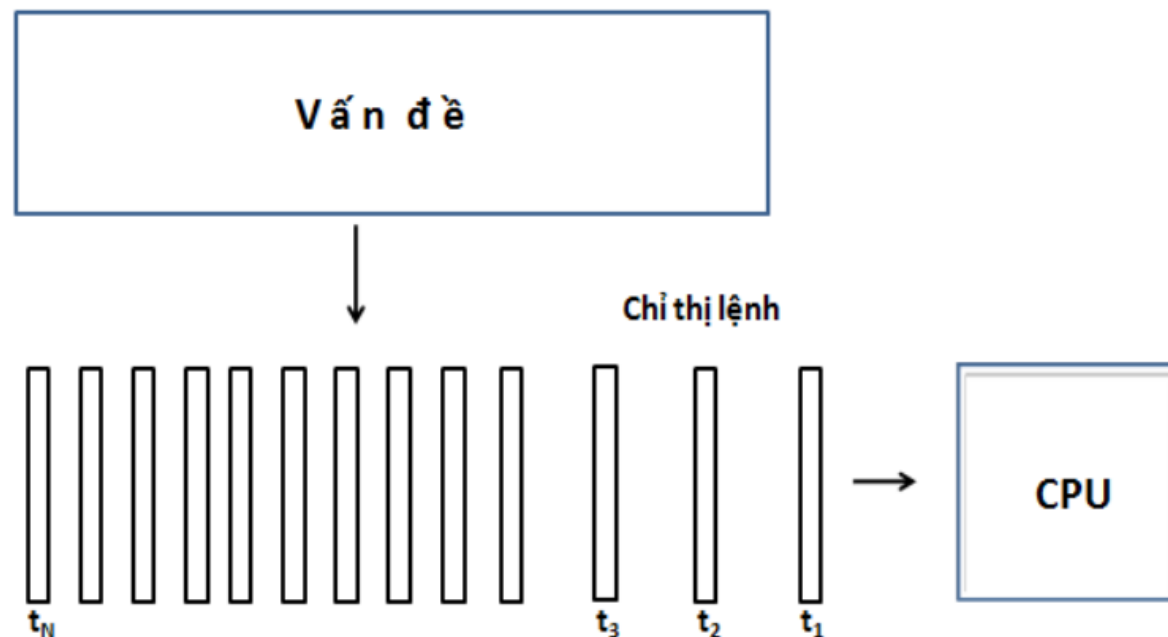
NĂNG LỰC HỆ THỐNG MÁY

- Distributed Computing
- Cluster Computing: bao gồm một số máy tính kết nối lỏng lẻo với nhau, làm việc cùng nhau, ở nhiều khía cạnh, có thể xem như một hệ thống duy nhất. Không giống như điện toán lưới, trong điện toán cụm, mỗi nút (node) thực hiện cùng một công việc giống nhau, được điều khiển và lập lịch bằng phần mềm
- Peer to peer Computing: peer to peer (P2P) là một kiến trúc ứng dụng phân tán nhằm phân vùng nhiệm vụ hoặc khối lượng công việc giữa các peer. Các peer là những thiết bị tham gia trong ứng dụng có đặc quyền như nhau. Chúng tạo thành một mạng lưới các node ngang hàng.
- Ambiguous Computing

TÍNH TOÁN TUẦN TỰ

CÁC VẤN ĐỀ TRONG THỰC TIỄN

- Phức tạp, yêu cầu về tốc độ tính toán
- Tính toán tuần tự (1 CPU): cần nhiều thời gian



GIỚI HẠN CỦA TÍNH TOÁN TUẦN TỰ

- Tốc độ truyền: phụ thuộc trực tiếp vào việc dữ liệu di chuyển thông qua các phần cứng
- Giới hạn trong việc thu nhỏ: công nghệ vi xử lý giúp tăng số transistor đặt trên 1 con chip. Tuy nhiên, khả năng đạt được còn gặp nhiều khó khăn
- Giới hạn về kinh tế: một bộ xử lý đơn mạnh giá cao hơn nhiều so với việc sử dụng nhiều bộ xử lý với giá vừa phải

GIỚI HẠN CỦA TÍNH TOÁN TUẦN TỰ

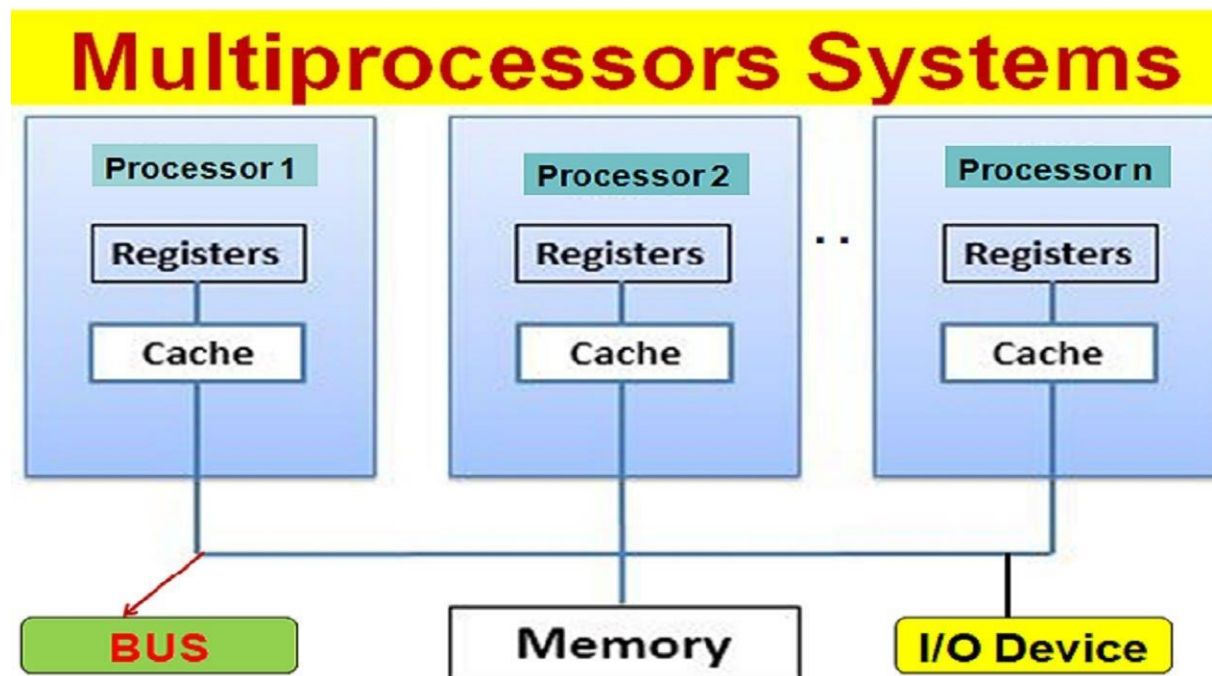
➤ Kiến trúc máy tính hiện đại cho phép có thể nâng cao năng lực xử lý:

- Nhiều bộ phận thực thi (Multiple execution units)
- Kiến trúc lệnh đường ống (Pipeline instructions): cho phép nạp một lệnh mới trong khi CPU đang xử lý lệnh trước đó.
- Multi-core

Chuỗi lệnh	Chu kỳ xung nhịp								
	1	2	3	4	5	6	7	8	9
Lệnh thứ i	IF	ID	EX	MEM	RS				
Lệnh thứ i+1		IF	ID	EX	MEM	RS			
Lệnh thứ i+2			IF	ID	EX	MEM	RS		
Lệnh thứ i+3				IF	ID	EX	MEM	RS	
Lệnh thứ i+4					IF	ID	EX	MEM	RS

TÍNH TOÁN SONG SONG (PARALLEL COMPUTING)

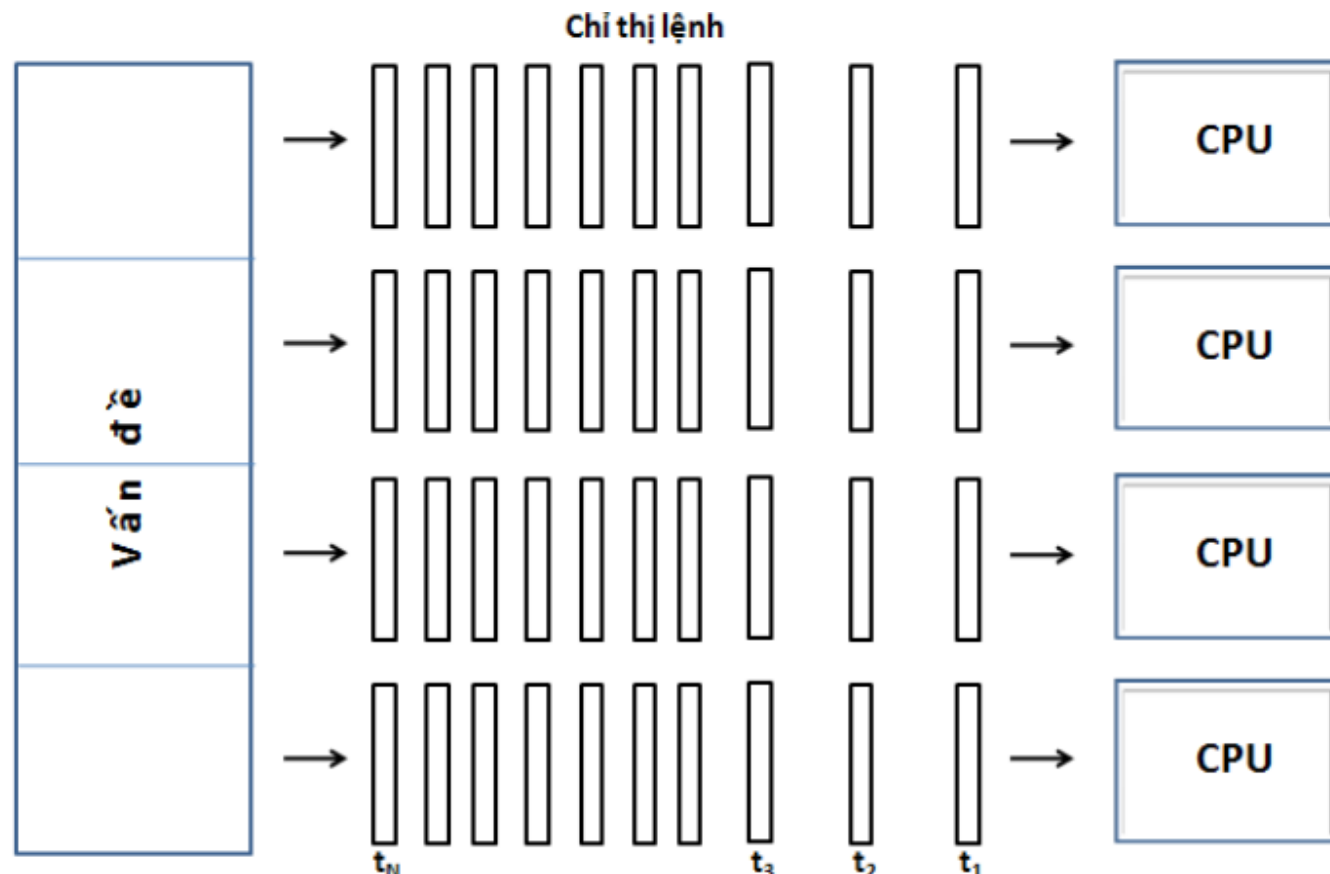
Để giải quyết vấn đề đặt ra, thay vì tăng tốc độ tính toán của máy tính (đòi hỏi thực hiện trong nhiều năm) → sử dụng cùng một lúc nhiều bộ xử lý (Multi-Processor)



TÍNH TOÁN SONG SONG (PARALLEL COMPUTING)

TÍNH TOÁN SONG SONG TRÊN NHIỀU CPU

- Vấn đề được chia thành nhiều phần *tương đối độc lập* và được thực hiện **đồng thời** → tăng tốc độ tính toán



TÍNH TOÁN SONG SONG (PARALLEL COMPUTING)

TÍNH TOÁN SONG SONG

- Là một sự tiến hóa của tính toán tuần tự nhằm cố gắng mô phỏng những vấn đề đặt ra trong thế giới tự nhiên chứa đựng:
 - Nhiều sự phức tạp
 - Các sự kiện liên quan với nhau xảy ra cùng một thời gian nhưng vẫn trong cùng một chuỗi

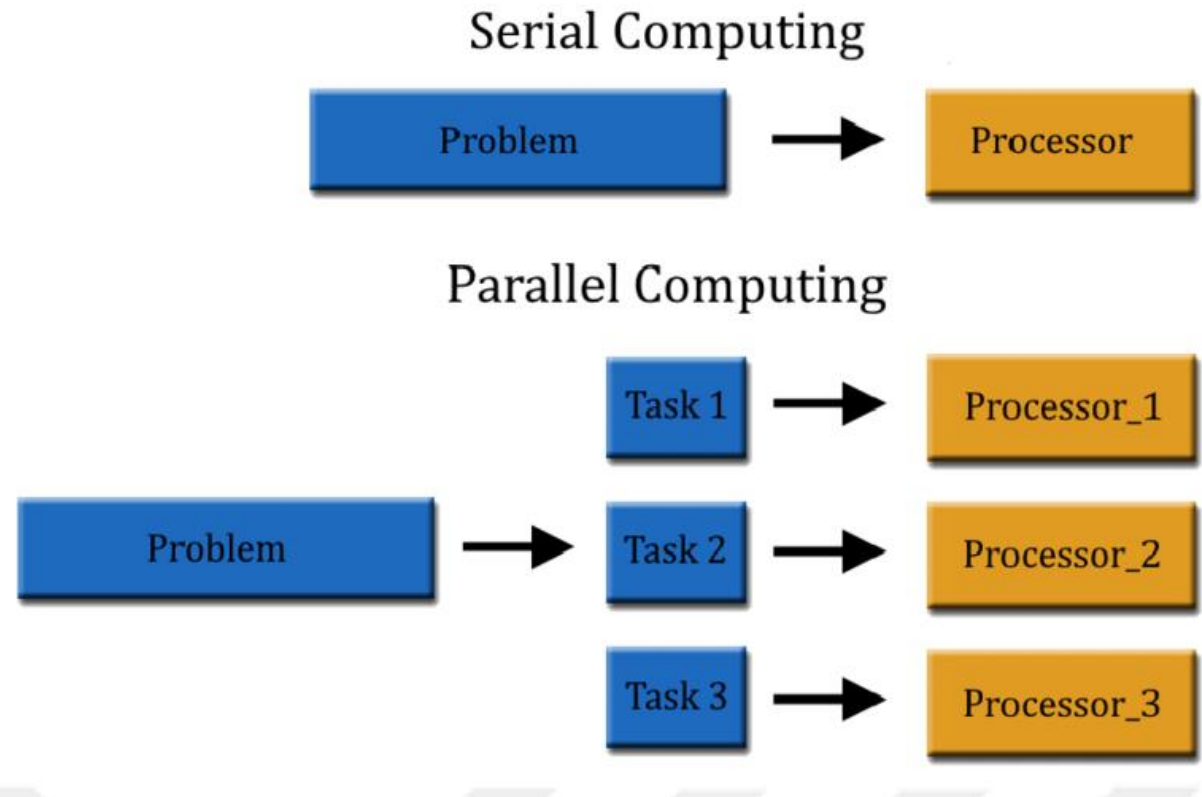
TÍNH TOÁN SONG SONG (PARALLEL COMPUTING)

TRONG MỘT NGŨ CẢNH ĐƠN GIẢN

- Tính toán song song là việc sử dụng đồng thời nhiều nguồn tài nguyên tính toán để giải quyết một vấn đề:
 - Được chạy bằng cách sử dụng nhiều CPU
 - Một vấn đề có thể chia ra thành nhiều phần rời rạc có thể giải quyết một cách đồng thời
 - Mỗi phần được tiếp tục phân chia thành một chuỗi các câu lệnh.
 - Những câu lệnh của mỗi phần được thi hành một cách đồng thời trên các CPU khác nhau.

TÍNH TOÁN SONG SONG (PARALLEL COMPUTING)

- Quá trình tính toán gồm nhiều tiến trình được kích hoạt đồng thời
- Các tiến trình cùng tham gia giải quyết một bài toán/ vấn đề trên hệ thống có nhiều bộ xử lý.



TÍNH TOÁN SONG SONG (PARALLEL COMPUTING)

NGUỒN TÀI NGUYÊN TÍNH TOÁN CÓ THỂ LÀ

- Một máy tính duy nhất với nhiều bộ xử lý
- Một số tùy ý máy tính được kết nối qua mạng
- Tổ hợp cả hai loại trên

6. Designing Parallel Programs

1. Automatic vs. Manual Parallelization
2. Understand the Problem and the Program
3. Partitioning
4. Communications
5. Synchronization
6. Data Dependencies
7. Load Balancing
8. Granularity
9. I/O
10. Limits and Costs of Parallel Programming
11. Performance Analysis and Tuning

7. Parallel Examples

1. Array Processing
2. PI Calculation
3. Simple Heat Equation
4. 1-D Wave Equation

8. References and More Information

- The compute resources might be:
 - A single computer with multiple processors;
 - An arbitrary number of computers connected by a network;
 - A combination of both.
- The computational problem should be able to:
 - Be broken apart into discrete pieces of work that can be solved simultaneously;
 - Execute multiple program instructions at any moment in time;
 - Be solved in less time with multiple compute resources than with a single compute resource.

TÍNH TOÁN TUẦN TỰ - SONG SONG

Tuần tự

```
void quicksort(int * a, int n)
{
    if (n <= 1) return;
    int s = partition(a,n);

    quicksort(a,s);
    quicksort(a+s,n-s);
}
```



Song song

```
void quicksort(int * a, int n)
{
    if (n <= 1) return;
    int s = partition(a,n);
    parallel_invoke(
        [&]{quicksort(a,s);},
        [&]{quicksort(a+s,n-s);});
}
```


TÍNH TOÁN TUẦN TỰ - SONG SONG

Tuần tự

```
void quicksort(int * a, int n)
{
    if (n <= 1) return;
    int s = partition(a,n);
    1 quicksort(a,s);
    2 quicksort(a+s,n-s);
}
```

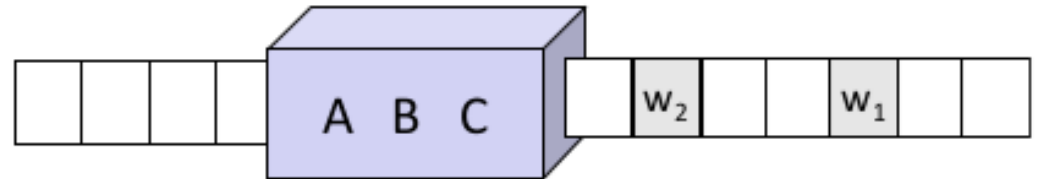
Song song

```
void quicksort(int * a, int n)
{
    if (n <= 1) return;
    int s = partition(a,n);
    3 parallel_invoke(
        [&]{quicksort(a,s);},
        [&]{quicksort(a+s,n-s);});
}
```

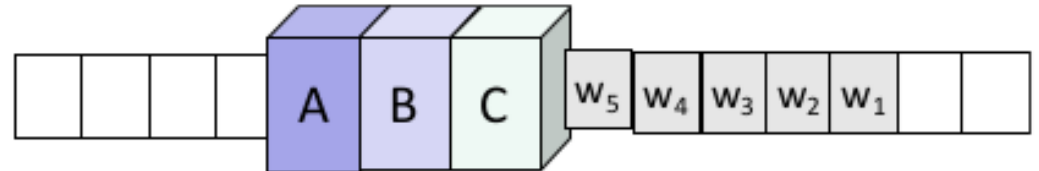
- Thời điểm 1 & 2 → biết **chắc** tình trạng của mảng a
- Thời điểm 3 → **không biết chắc** tình trạng của mảng a

TÍNH TOÁN TUẦN TỰ - SONG SONG

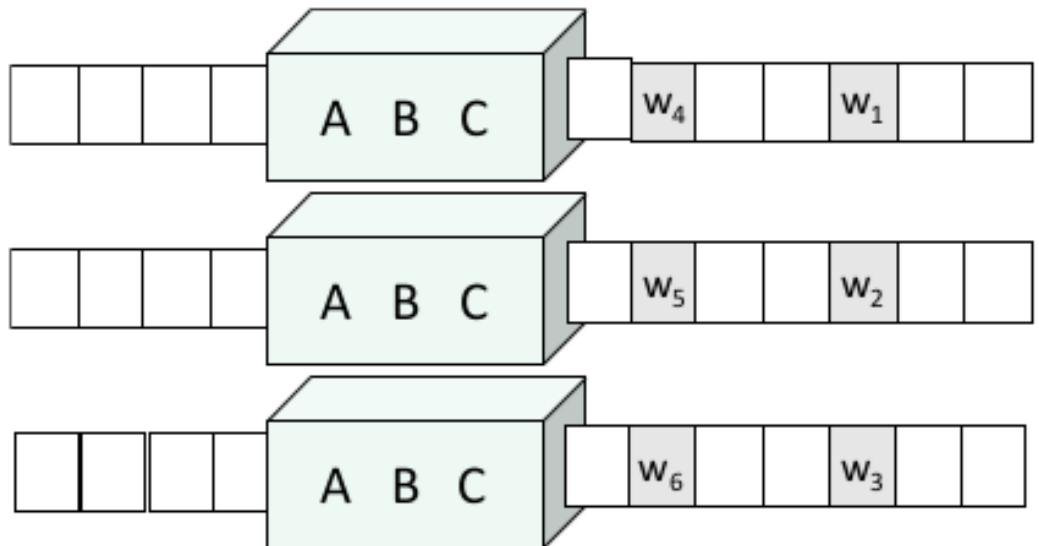
Sequential



Pipeline



Data parallelism



XỬ LÝ SONG SONG TRONG CUỘC SỐNG

CÁC VẤN ĐỀ ĐƯỢC XỬ LÝ SONG SONG TRONG CUỘC SỐNG



XỬ LÝ SONG SONG TRONG CUỘC SỐNG

- Xử lý song song hoàn toàn không xa lạ trong cuộc sống
 - Quầy tính tiền ở siêu thị
 - Mua vé vào công viên
 - Đường cao tốc nhiều làn xe
- Nhiều sự việc phức tạp trong cuộc sống đều xảy ra đồng thời

TẠI SAO TÍNH TOÁN SONG SONG?

YÊU CẦU NGƯỜI DÙNG

- Cần thực hiện một khối lượng công việc lớn
- Thời gian xử lý nhanh



TẠI SAO TÍNH TOÁN SONG SONG?

YÊU CẦU THỰC TIỄN

- Không thể tạo ra một máy tính với tài nguyên bộ nhớ và khả năng tính toán vô hạn.
- Có nhiều bài toán mà xử lý tuần tự không đáp ứng được.
- Việc tính toán trên hệ thống có nhiều CPU nhanh hơn trên hệ thống có 1 CPU.
- Tính toán song song sẽ giải quyết được bài toán lớn, phức tạp.

TÀI NGUYÊN TÍNH TOÁN SONG SONG

- Một máy tính đơn có nhiều bộ xử lý
- Nhiều máy tính được kết nối mạng
- Một siêu máy tính có nhiều bộ đa xử lý
- Phần cứng chuyên biệt
 - Field - Programmable Gate Array (FPGA)
 - Bộ xử lý đa năng của card đồ họa (GPU)
- Hoặc có thể là sự kết hợp của các loại thiết bị như trên

HÌNH THỨC TÍNH TOÁN SONG SONG

TÍNH TOÁN SONG SONG:

- Song song cấp bit
- Song song cấp lệnh (ngôn ngữ lập trình)
- Song song dữ liệu
- Song song tác vụ (chương trình con)

MỘT SỐ KHÁI NIỆM

LANTENCY VÀ BANDWIDTH

- Tiêu chuẩn để xác định hiệu năng của một hệ thống máy tính
- Lantency: thời gian chờ để nhận được dữ liệu khi dữ liệu đã chuyển đi (khoảng thời gian giữa lúc yêu cầu được gửi đi - request sent và lúc kết quả được trả về - response received)
- Bandwidth: tốc độ mà dữ liệu có thể đưa vào từ bộ nhớ đến bộ xử lý

MỘT SỐ KHÁI NIỆM

- Bandwidth cho biết số lượng phép tính thực hiện được, Lantency là thời gian bị trễ → Lantency sẽ làm ảnh hưởng đến hệ thống.
- Ví dụ:
 - Giả sử vòi phun nước sẽ phun nước sau 5s kể từ khi mở van → Lantency là 5s.
 - Nếu vòi phun với tốc độ 2 gallon/s → Bandwidth là 2 gallon/s

MỘT SỐ KHÁI NIỆM

- Trên hệ thống có Lantency là 100 ns, biết mỗi phép toán đòi hỏi phải nạp 1 dữ liệu vào bộ nhớ \rightarrow tốc độ tối đa để tính tích vô hướng của 2 vector?



MỘT SỐ KHÁI NIỆM

- Trên hệ thống có Lantency là 100 ns, biết mỗi phép toán đòi hỏi phải nạp 1 dữ liệu vào bộ nhớ
- → tốc độ tối đa để tính tích vô hướng của 2 vector bị giới hạn bởi 1 phép tính trên mỗi 100 ns = $10^2 \times 10^{-9} = 10^{-7}$ s

MỘT SỐ KHÁI NIỆM

- Trên hệ thống có Lantency là 100 ns, biết mỗi phép toán đòi hỏi phải nạp 1 dữ liệu vào bộ nhớ
- → tốc độ tối đa để tính tích vô hướng của 2 vector bị giới hạn bởi 1 phép tính trên mỗi $100 \text{ ns} = 10^2 \times 10^{-9} = 10^{-7} \text{ s}$
- → Giả sử máy tính bị giới hạn bởi 1 giây → thực hiện bao nhiêu phép tính?



MỘT SỐ KHÁI NIỆM

- Trên hệ thống có Lantency là 100 ns, biết mỗi phép toán đòi hỏi phải nạp 1 dữ liệu vào bộ nhớ
- → tốc độ tối đa để tính tích vô hướng của 2 vector bị giới hạn bởi 1 phép tính trên mỗi $100 \text{ ns} = 10^2 \times 10^{-9} = 10^{-7} \text{ s}$
- → Giả sử máy tính bị giới hạn bởi 1 giây → thực hiện được $1/10^{-7}$ phép tính = 10 MFlop/s

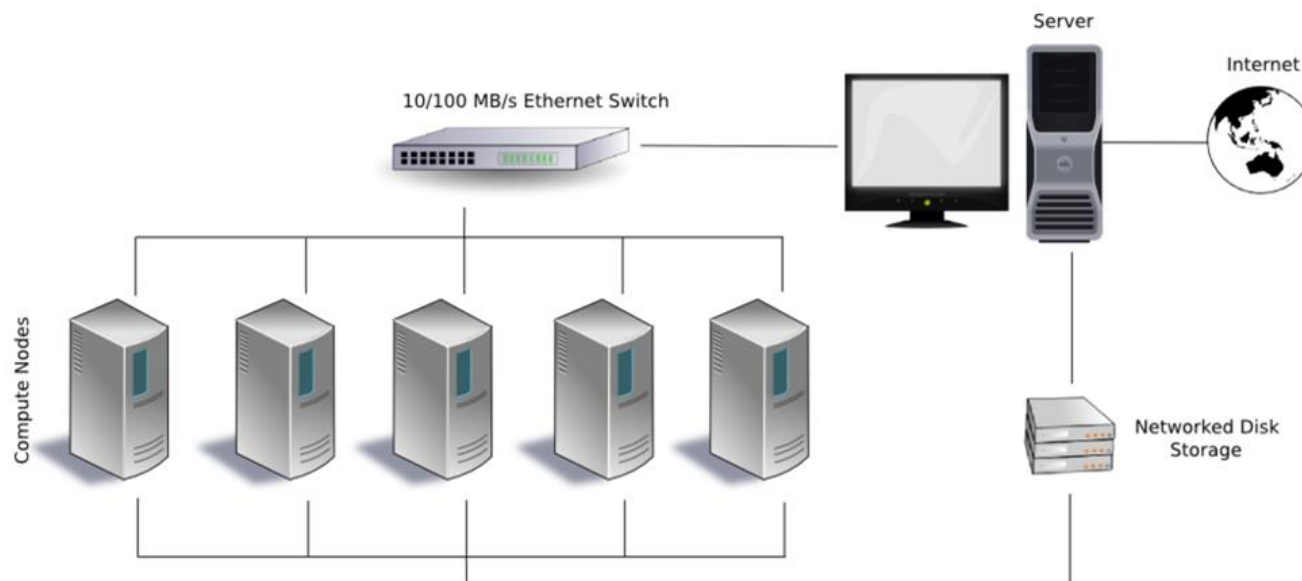
PC CLUSTER



PC CLUSTER

PC CLUSTER

- Nhóm các nút tính toán (chẳng hạn như PC)
- Mỗi nút chạy hệ điều hành của riêng nó
- Được kết nối cục bộ (tốc độ cao)
- Được xem như một “siêu” máy tính đơn

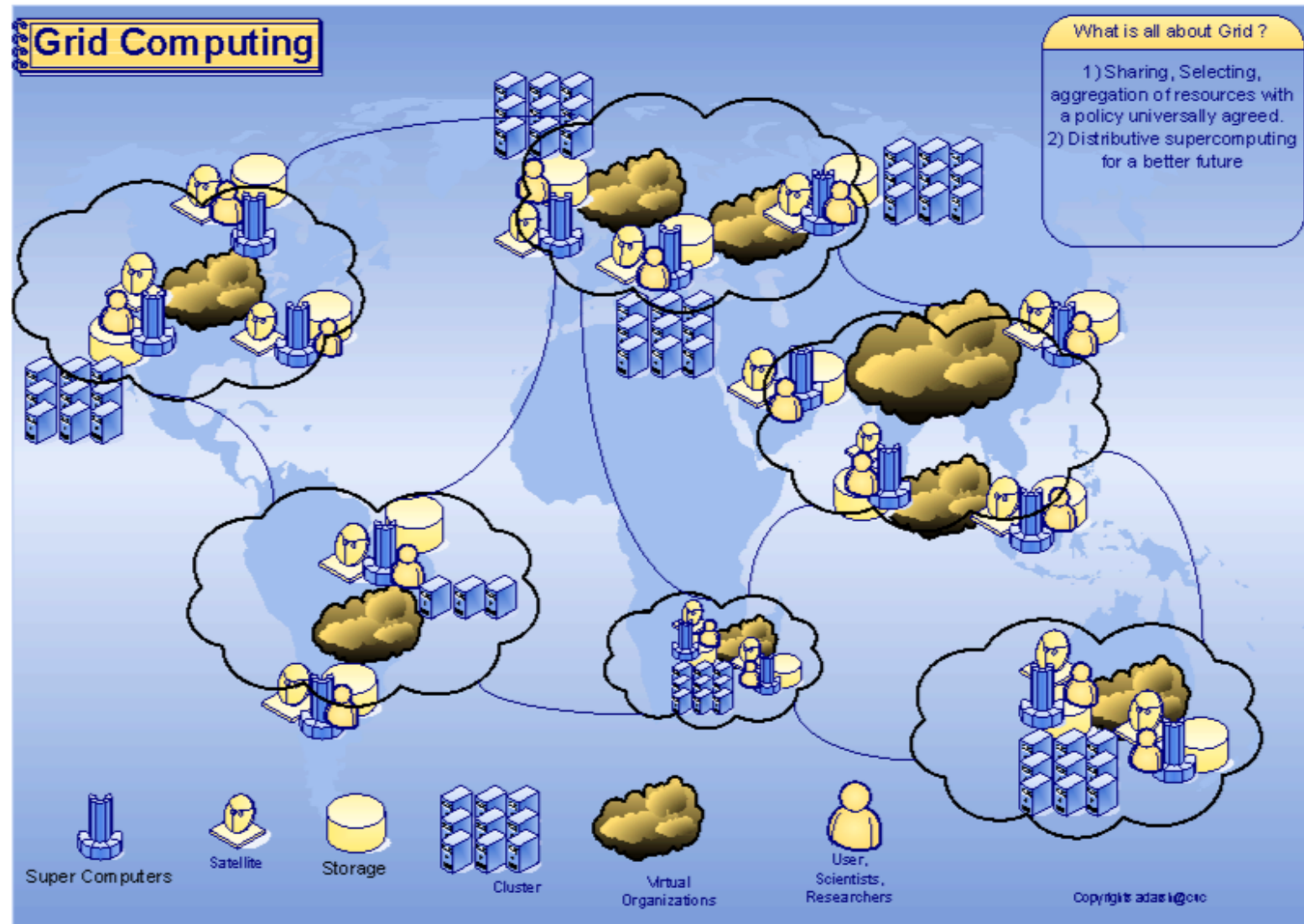


PC CLUSTER

ĐẶC TRƯNG CỦA PC CLUSTER

- Sử dụng microprocessor: chi phí thấp
- Mạng tốc độ cao
- Khả năng chịu đựng hỏng hóc
- Phần mềm tính toán phân tán với hiệu năng cao
- Giao diện MPI: quản lý tính toán trên cluster

LƯỚI (GRID)



LƯỚI (GRID)

GIỚI THIỆU VỀ LƯỚI

- Tập hợp các tài nguyên đa dạng
- Siêu máy tính, PC clusters, hệ thống lưu trữ
- Được nối kết tốc độ cao
- Chia sẻ
- Phân tán
- Đa dạng: hệ điều hành, hệ thống tập tin
- Được xem như một “siêu” máy tính đơn
- Khả năng chịu đựng hỏng hóc
- Phần mềm tính toán phân tán với hiệu năng cao

LƯỚI (GRID)

KHÁI NIỆM

- Ứng dụng: nhiều công việc thực thi song song trên nhiều máy tính khác nhau của lưới
- Đăng ký sử dụng tài nguyên
- Lập lịch: người dùng đề xuất hay hệ thống tự tìm tài nguyên thích hợp
- Hệ thống phân công việc đến các tài nguyên như lịch đã lập
- Hệ thống quản lý: theo dõi tài nguyên sẵn dùng, trạng thái hiện tại của công việc, người dùng
- An ninh: chứng thực, xác thực, đăng nhập

LƯỚI (GRID)

LỢI ÍCH

- Tiết kiệm thời gian và tài nguyên
- Thực thi một chương trình song song trên nhiều máy tính khác nhau
- Tận dụng tài nguyên nhàn rỗi trên mạng: CPU, đĩa
- Tài nguyên ảo: tận dụng tất cả tài nguyên không đồng nhất
- Cân bằng tải sử dụng tài nguyên
- Độ tin cậy: khả năng chịu đựng hỏng hóc
- Chi phí thấp
- Dễ bảo trì, quản lý

LƯỚI Ý

- Yêu cầu tính toán
- Xu hướng phát triển máy tính
- Tính toán tuần tự
- Tính toán song song
- PC Cluster



