

# Tính toán song song và phân tán

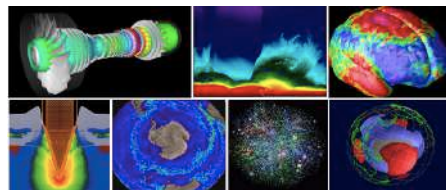
PGS.TS. Trần Văn Lăng

langtv@vast.vn

Tài liệu: Introduction to Parallel Computing

Blaise Barney, Lawrence Livermore National Laboratory

[https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)



Introduction to Parallel Computing

John R. Barst, Lawrence Livermore National Laboratory

Table of Contents

1. Abstract
2. Overview
3. Concepts and Terminology
4. Parallel Computing Memory Architectures
5. Shared Memory
6. Distributed Memory
7. Distributed Shared Memory
8. Hybrid Shared/Shared Memory
9. Hybrid Shared/Shared Memory
10. Hybrid Shared/Shared Memory
11. Hybrid Shared/Shared Memory
12. Hybrid Shared/Shared Memory
13. Hybrid Shared/Shared Memory
14. Hybrid Shared/Shared Memory
15. Hybrid Shared/Shared Memory
16. Hybrid Shared/Shared Memory
17. Hybrid Shared/Shared Memory
18. Hybrid Shared/Shared Memory
19. Hybrid Shared/Shared Memory
20. Hybrid Shared/Shared Memory

Dr. Tran Van Lang, Assoc. Prof. in Computer Science

1

## Table of Contents

1. Abstract
2. Overview
3. Concepts and Terminology
4. Parallel Computing Memory Architectures
5. Shared Memory
6. Distributed Memory
7. Distributed Shared Memory
8. Hybrid Shared/Shared Memory
9. Hybrid Shared/Shared Memory
10. Hybrid Shared/Shared Memory
11. Hybrid Shared/Shared Memory
12. Hybrid Shared/Shared Memory
13. Hybrid Shared/Shared Memory
14. Hybrid Shared/Shared Memory
15. Hybrid Shared/Shared Memory
16. Hybrid Shared/Shared Memory
17. Hybrid Shared/Shared Memory
18. Hybrid Shared/Shared Memory
19. Hybrid Shared/Shared Memory
20. Hybrid Shared/Shared Memory

## Concepts and Terminology

### von Neumann Architecture

- Named after the Hungarian mathematician John von Neumann

## 2. Khái niệm và thuật ngữ

1. Kiến trúc máy tính Von Neumann
2. [Phân loại kinh điển của Flynn](#)
3. [Một vài thuật ngữ song song](#)

Dr. Tran Van Lang, Assoc. Prof. in Computer Science

2

## Kiến trúc máy tính Von Neumann

- Được đặt theo tên của Nhà toán học người Hungary - John von Neumann – người đầu tiên đưa ra những yêu cầu của một máy tính điện tử (electronic computer) trong công trình của Ông vào năm 1945.



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

3

- Từ đó, hầu như tất cả các máy tính đều tuân theo thiết kế cơ bản này.
- Sự khác nhau giữa các máy chỉ là sự sắp đặt của các hệ thống dây kết nối cứng (hard wiring).

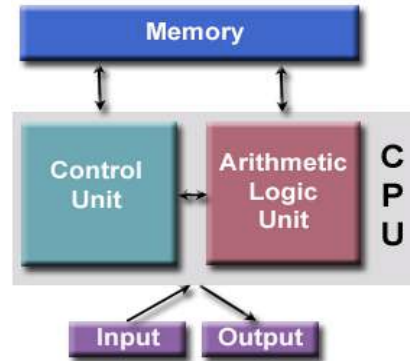


Dr. Tran Van Lang, Assoc. Prof. in Computer Science

4

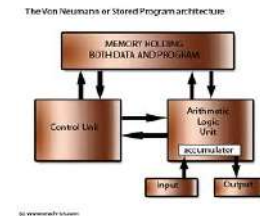
- Theo Von Neumann, một máy tính điện tử bao gồm 4 thành phần chính:

- Memory
- Control Unit
- Arithmetic Logic Unit
- Input/Output

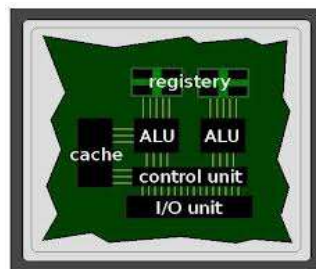


- Memory: bộ nhớ đọc ghi ngẫu nhiên lưu trữ các lệnh chương trình và dữ liệu khi thực thi

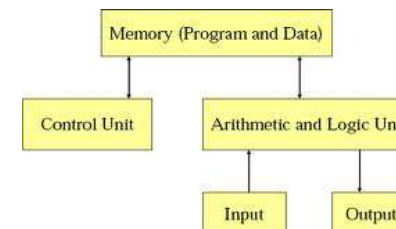
- Lệnh chương trình (Program Instruction) được mã hóa dữ liệu để yêu cầu máy tính làm điều gì đó
- Còn dữ liệu (Data) đơn giản chỉ là thông tin được sử dụng bởi chương trình



- Control Unit: lấy các lệnh chương trình và dữ liệu từ bộ nhớ, giải mã câu lệnh rồi phối hợp một cách tuần tự các phép toán, các thao tác để hoàn thành nhiệm vụ được lập trình sẵn.



- Arithmetic Logic Unit: thực hiện các phép toán số học và luận lý cơ bản.
- Input/Output: nhằm giao tiếp với người sử dụng hoặc thiết bị ngoại vi



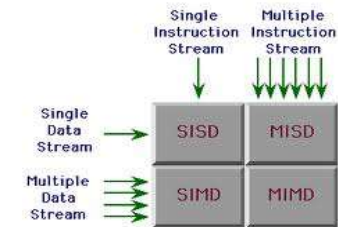
## Kiến trúc song song

- Máy tính song song vẫn giữ nguyên thiết kế cơ bản này; chỉ nhân số bộ phận (Unit) lên nhiều lần.
- Về cơ bản, kiến trúc nền tảng còn lại như kiến trúc máy tính tuần tự.

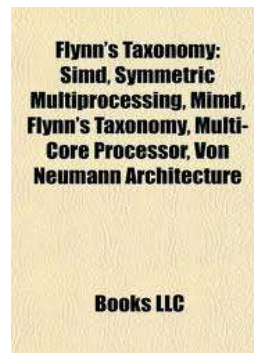


## Phân loại theo Flynn

- Có nhiều cách để phân loại máy tính song song. Một trong những phân loại được sử dụng rộng rãi có từ năm 1966, đó là phân loại Flynn.



- Flynn phân biệt kiến trúc máy tính song song dựa trên số trạng thái có thể của câu lệnh và dữ liệu.
- Số trạng thái ở đây là Single hay Multiple.

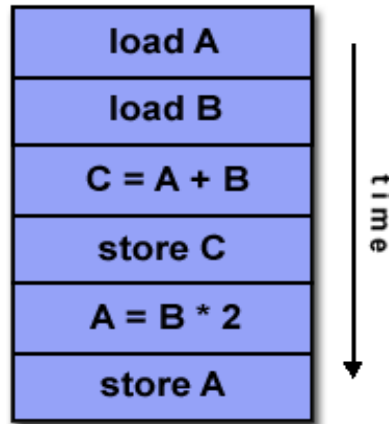


## Ma trận phân loại theo Flynn

<b>SISD</b> Single Instruction, Single Data	<b>SIMD</b> Single Instruction, Multiple Data
<b>MISD</b> Multiple Instruction, Single Data	<b>MIMD</b> Multiple Instruction, Multiple Data

# Single Instruction, Single Data (SISD)

- Là một máy tính tuần tự (non-parallel)
- **Single Instruction:** Chỉ một dòng câu lệnh được tác động bởi CPU trong suốt một chu kỳ đồng hồ.
- **Single Data:** Chỉ một dòng dữ liệu được dùng như đầu vào trong suốt một chu kỳ đồng hồ.



# Single Instruction, Single Data

- Đây là một loại máy tính lâu đời nhất, đồng thời là loại máy tính phổ biến nhất trong thời đại ngày nay, chẳng hạn:
  - máy tính lớn thế hệ cũ,
  - máy mini,
  - máy trạm và
  - hầu hết các máy tính PC hiện đại ngày nay.



CDC 7600

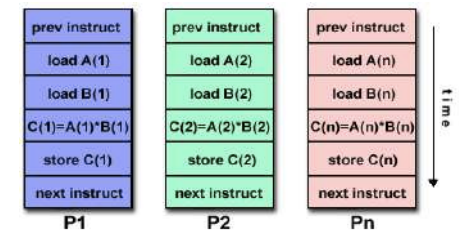


PDP1



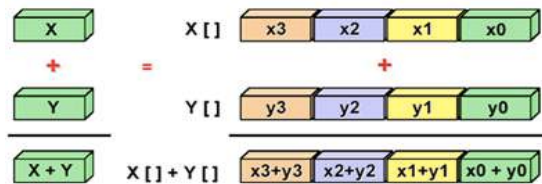
# Single Instruction, Multiple Data (SIMD)

- Loại máy tính song song
- **Single Instruction:** Tất cả các đơn vị xử lý thi hành cùng câu lệnh ở một chu kỳ đồng hồ cho trước.
- **Multiple Data:** Mỗi đơn vị xử lý có thể thực hiện thao tác trên phần tử dữ liệu khác nhau.



Dữ liệu A, B, C khác nhau trên các đơn vị xử lý P

- Loại tốt nhất phù hợp cho những bài toán chuyên môn đòi hỏi tốc độ xử lý cao như xử lý đồ thị, hình ảnh.
- Thi hành một cách đồng bộ
- Ví dụ, với câu lệnh  $x + y$

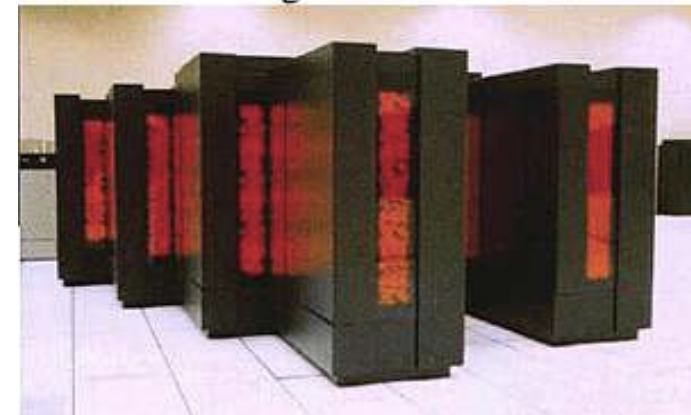


- Có hai biến thể của loại SIMD này:
  - Processor Arrays: Connection Machine CM-2, MasPar MP-1 & MP-2, ILLIAC IV
  - Vector Pipelines: IBM 9000, Cray X-MP, Y-MP & C90, Fujitsu VP, NEC SX-2, Hitachi S820, ETA10
- Hầu hết các máy tính hiện đại, đặc biệt các Graphics Processor Units (GPUs) sử dụng chỉ thị SIMD.

## Processor Arrays



## Thinking Machines CM-2



# Vector Pipelines

IBM 9000



Cray X-MP



Cray Y-MP



NEC - SX



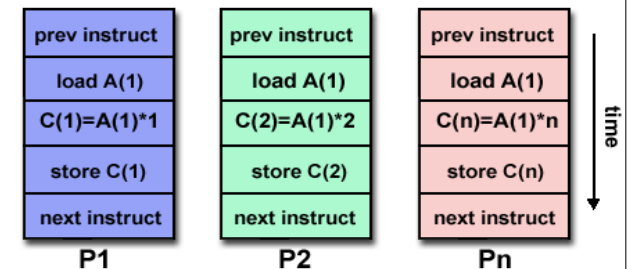
Fujitsu VP



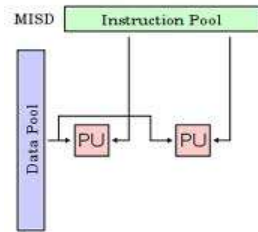
## Multiple Instruction, Single Data (MISD)

- Loại máy tính song song
- **Multiple Instruction:** Mỗi đơn vị xử lý hoạt động trên dữ liệu độc lập thông qua dòng câu lệnh riêng biệt.
- **Single Data:** Mỗi dòng dữ liệu duy nhất được đưa vào nhiều đơn vị xử lý.

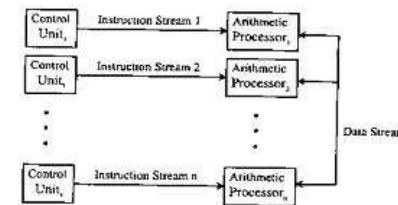
A(1) được đưa vào các P để tính ra các giá trị C khác nhau



- Đây là lớp phân loại để đầy đủ, không tồn tại loại máy tính này bao giờ.
- Tuy nhiên, có một vài dự án thử nghiệm. Một trong số đó là máy tính Carnegie-Mellon C.mmp (1971).



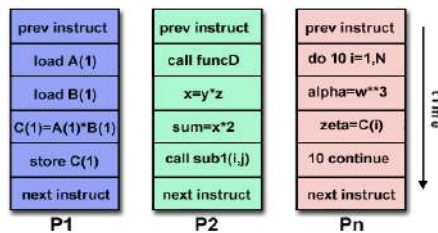
- Một vài ứng dụng có thể tương tượng:
  - Bộ lọc nhiễu tần số hoạt động trên một tín hiệu duy nhất
  - Nhiều thuật toán giải mã cố gắng crack một tin nhắn duy nhất được mã hóa.



## Multiple Instruction, Multiple Data (MIMD)

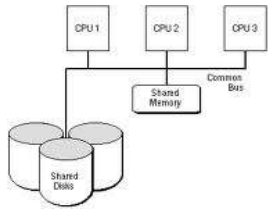
- Loại máy tính song song
- **Multiple Instruction:** Mỗi bộ xử lý có thể thi hành dòng câu lệnh khác nhau.
- **Multiple Data:** Mỗi bộ xử lý có thể làm việc với dòng dữ liệu khác nhau.

Mỗi bộ xử lý P thực hiện một số câu lệnh khác nhau – lệnh thứ 2, 3, 4, 5



- Việc thực thi có thể đồng bộ hoặc không đồng bộ, có thể xác định hoặc không xác định (non-deterministic)
- Hiện tại, loại máy tính song song hiện đại nhất, phổ biến nhất thuộc vào loại này. Chẳng hạn:
  - Hầu hết các siêu máy tính hiện tại,
  - Cụm và lưới (clusters and grids) máy tính song song nối mạng,

- Máy tính đa xử lý đối xứng (multi-processor SMP)
- multi-core PCs.
- Lưu ý: Nhiều kiến trúc MIMD cũng bao gồm các thành phần con thi hành dạng SIMD.



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

29

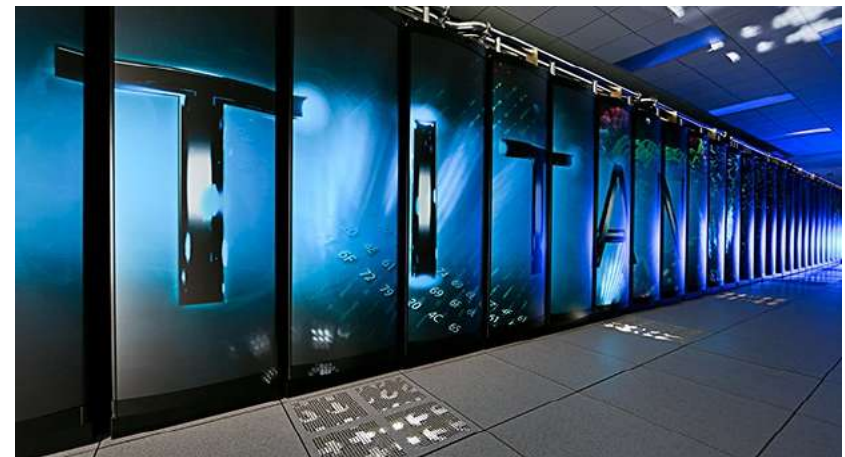
## K Computer, siêu máy tính của 11/2011



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

30

## Cray's Jaguar supercomputer upgraded with NVIDIA Tesla GPUs, renamed Titan (Oct 29<sup>th</sup> 2012)



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

32



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

31



## Một vài thuật ngữ

- **Supercomputing / High Performance Computing (HPC)**

- Lĩnh vực sử dụng máy tính nhanh và lớn để giải quyết những bài toán lớn.
- Thiết kế thuật toán để xây dựng chương trình tính toán song song



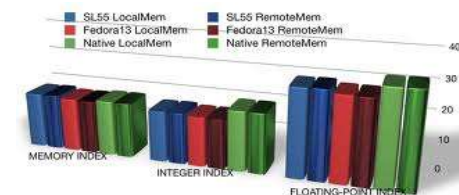
- **Node**

- Có thể coi như là máy tính nằm trong một chiếc hộp độc lập (stand-alone)
- Thông thường bao gồm nhiều CPUs/processors/cores.
- Node cũng có thể là một mạng (các máy nối mạng cùng nhau để bao gồm một siêu máy).

- **CPU / Socket / Processor / Core**

- Đây là những biến thể khác nhau, tùy thuộc vào ngữ cảnh trong cuộc nói chuyện.
- Trong quá khứ, CPU (Central Processing Unit) là một thành phần thực thi duy nhất từ một máy tính.
- Sau đó, nhiều CPU được tích hợp vào một Node.

- Rồi một CPU riêng lẻ được phân thành nhiều Core – là một đơn vị thực thi duy nhất.
- CPU với nhiều Core thỉnh thoảng được gọi là Socket – tùy theo nhà sản xuất.
- Kết quả là một Node với nhiều CPU, mỗi CPU chứa nhiều Core.





Supercomputer - each blue light is a node  
 Node - standalone Von Neumann computer  
 CPU / Processor / Socket - each has multiple cores / processors.



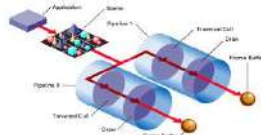
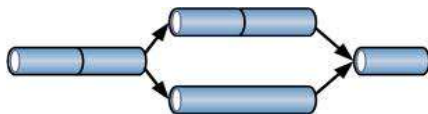
## • Task

- Là một công việc giống như một chương trình hoặc một đoạn chương trình tuần tự thực thi trên một bộ xử lý.
- Một chương trình song song bao gồm nhiều task chạy trên nhiều bộ xử lý.



## • Pipelining

- Tách một task thành ra nhiều bước để thực thi trên nhiều đơn vị xử lý khác nhau.
- Khi đó, mỗi đơn vị xử lý có thể nạp một bước mới vào để thực hiện trong khi đang thực hiện một bước trước đó giống như một dây chuyền lắp ráp.
- Đây là một loại tính toán song song

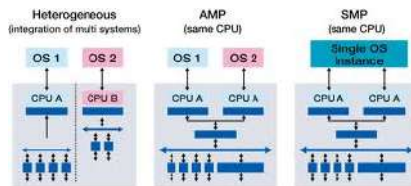


## • Shared Memory

- Mô tả kiến trúc máy tính mà tất cả các bộ xử lý truy cập một cách trực tiếp đến một bộ nhớ vật lý chung.
- Trong ngữ cảnh lập trình, bộ nhớ chia sẻ dùng để mô tả mô hình mà những task song song có cùng "bức tranh" của bộ nhớ. Những task này có thể truy cập một cách trực tiếp đến các biến trong bộ nhớ mà không cần biết sự tồn tại thực sự của bộ nhớ vật lý ở đâu.

## • Symmetric Multi-Processor (SMP)

- Kiến trúc phần cứng ở đó nhiều bộ xử lý chia sẻ một không gian địa chỉ chung (global address space) và truy cập đến tất cả nguồn tài nguyên
- Có thể nói đây là kiến trúc để thực hiện tính toán với bộ nhớ chia sẻ.



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

41

## • Distributed Memory

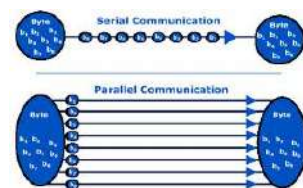
- Về mặt phần cứng, việc truy cập bộ nhớ vật lý của máy tính mạng thì không thể dùng chung cho các máy trong mạng đó.
- Với mô hình lập trình, một task chỉ có thể “thấy được” vùng bộ nhớ địa phương mà trên đó nó thực hiện. Nên khi task này muốn truy cập một vùng bộ nhớ trên máy khác, thì phải thông qua sự giao tiếp.

Dr. Tran Van Lang, Assoc. Prof. in Computer Science

42

## • Communications

- Những task song song thường cần phải trao đổi dữ liệu.
- Có một vài cách để điều này có thể hoàn thành như thông qua bus bộ nhớ chia sẻ hay truyền thông điệp trên một mạng máy tính

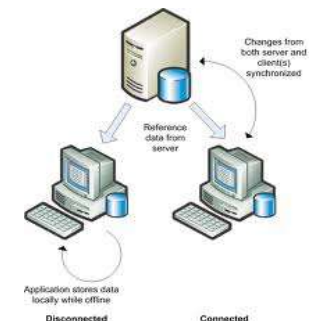


Dr. Tran Van Lang, Assoc. Prof. in Computer Science

43

## • Synchronization

- Sự phối hợp để thực hiện các task song song theo thời gian thực, trong đó việc giao tiếp được diễn ra một cách thường xuyên
- Sự đồng bộ hóa thường liên quan đến việc chờ đợi ít nhất là một task khác, nên thời gian thực thi song song bị kéo dài thêm.



Dr. Tran Van Lang, Assoc. Prof. in Computer Science

44

## • Granularity

- Trong tính toán song song, granularity (độ chi tiết) là độ đo chất lượng của tỷ lệ giữa việc tính toán (computation) và việc giao tiếp (communication).
- Có 2 giá trị:
  - **Coarse (thô)**: một lượng công việc tính toán tương đối lớn được làm giữa các sự kiện giao tiếp.
  - **Fine (tinh)**: một lượng công việc tính toán tương đối nhỏ được làm giữa các sự kiện giao tiếp

## • Observed Speedup

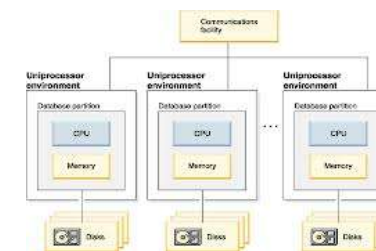
- Độ tăng tốc quan sát được (Observed Speedup) của một chương trình được song song,
- Định nghĩa như là tỷ số giữa thời gian thi hành tuần tự và thời gian thi hành song song
- Một trong những chỉ số đơn giản nhất và được sử dụng rộng rãi nhất khi quan tâm đến hiệu năng (Performance) của một chương trình song song.

## • Parallel Overhead

- Lượng thời gian cần thiết để phối hợp các task song song. Chi phí song song (Parallel overhead) có thể bao gồm các yếu tố chi phí như:
  - Thời gian khởi động task (start-up time)
  - Đồng bộ hóa
  - Truyền dữ liệu
  - Chi phí phần mềm bị áp đặt bởi trình biên dịch song song, các thư viện, các công cụ, hệ điều hành, v.v...
  - Thời gian kết thúc task (termination time)

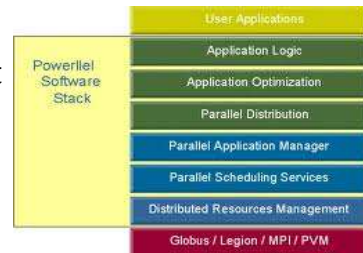
## • Massively Parallel

- Song song quy mô lớn đề cập đến yếu tố phần cứng bao gồm một hệ thống song song có nhiều bộ xử lý.



- **Embarrassingly Parallel**

- Giải quyết nhiều tác vụ tương tự nhưng độc lập một cách đồng thời mà không cần đến sự phối hợp giữa chúng.



- **Scalability**

- Khả năng mở rộng: đề cập đến khả năng (cả phần cứng lẫn phần mềm) của hệ thống song song, qua đó minh chứng sự tăng lên tương ứng của speedup khi bổ sung thêm nhiều bộ xử lý.

- Đôi khi còn được hiểu theo nghĩa khả năng tạo tỷ lệ giữa bộ xử lý và bộ nhớ để bảo đảm speedup

- Những yếu tố góp phần cho khả năng mở rộng:
  - Bandwidths và giao tiếp của mạng.
  - Thuật toán
  - Chi phí song song liên quan
  - Đặc điểm của chương trình