**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG PHÂN TÁN**

**1.1. Định nghĩa**

* ***Mục tiêu hệ thống phân tán:***
* Đảm bảo tính sẵn sàng của tài nguyên ***(quan trọng nhất)***
* Che giấu sự thật các tiến trình và tài nguyên phân tán trên nhiều máy tính
* Cho phép chia sẻ tài nguyên
* Qui mô hệ thống
* Lỗi xảy ra trên một máy chủ nào đó sẽ ảnh hưởng ít nhất đến vận hành chung của hệ thống
* ***Những thành tựu cơ bản cho phép chuyển từ hệ thống tập trung sang hệ thống phân tán***
* Tiến bộ thứ nhất: sự phát triển mạnh mẽ bộ vi xử lý cho máy tính cá nhân
* Tiến bộ thứ hai: sự phát minh ra mạng máy tính tốc độ cao
* ***Bắt nguồn từ nhu cầu:*** sử dụng khả năng tính toán tốt hơn và hiệu quả hơn
* ***3 yếu tố quyết định tốc độ tính toán:***
* Bộ vi xử lý trung tâm CPU
* Bộ nhớ RAM
* Kênh truyền trên bo mạch chủ
* ***Trong nhiều trường hợp, bắt buộc xây dựng hệ thống phân tán vì:***
* Yêu cầu tính toán phân toán
* Yêu cầu khả năng chịu lỗi:
* Chia sẻ tài nguyên
* ***Khó khăn khi triển khai hệ thống phân tán:***
* Có nhiều loại máy tính và thiết bị mạng khác nhau của nhiều nhà sản xuất khác nhau và các máy tính cài đặt hệ điều hành khác nhau
* Khó tích hợp các phần mềm vì chúng được phát triển trên các ngôn ngữ lập trình khác nhau
* Thời gian phát triển phần mềm lâu hơn, số lượng dòng lệnh và các thủ tục nhiều hơn => chi phí xây dựng cao

**1.2. Quan điểm phần cứng**

* ***Dựa trên mô hình OSI***
* ***Michael J.Flynn chỉ ra 2 đặc trưng cơ bản cho 1 hệ thống tính toán:***
* Số lượng luồng xử lý lệnh
* Số lượng kênh truyền dữ liệu

=> xây dựng được 4 loại

* ***Hệ thống nhiều bộ vi xử lý:*** các bộ vi xử lý dùng chung bộ nhớ, việc trao đổi thông tin giữa các bộ vi xử lý được thực hiện bằng cách đọc/ghi các ô nhớ.
* ***Hệ thống nhiều máy tính:*** giảm thiểu xung đột truyền thông cho các máy tính trong mạng cục bộ

Giái pháp: Đấu nối các máy tính theo hình sao sử dụng thiết bị chuyển mạch

**1.3. Quan điểm về phần mềm**

| ***Hệ thống*** | ***Mô tả*** | ***Mục tiêu*** |
| --- | --- | --- |
| ***HĐH phân tán*** | Liên kết chặt dùng cho các hệ thống máy tính thuần nhất  LOCUS 1980 | Che giấu và quản lý các tài nguyên phần cứng |
| ***HĐH mạng*** | Liên kết lỏng, dùng cho các máy tính không thuần nhất  Microsoft Server, Linux, Unix | Cung cấp dịch vụ cục bộ cho các máy tính khác truy nhập từ xa |
| ***Phần mềm trung gian*** | Tầng trên của hệ điều hành mạng, cài đặt các dịch vụ mục đích chung | Cung cấp tính trong suốt  cho hệ thống phân tán |

* ***Hệ điều hành cho nhiều bộ xử lý:***
* ***Đa xử lý đối xứng:***
* Bộ xử lý nào cũng có thể chạy 1 loại tiến trình bất kỳ, giao tiếp với nhau thông qua 1 bộ nhớ dùng chung
* Cung cấp cơ chế chịu lỗi và khả năng cân bằng tải tối ưu hơn
* ***Đa xử lý bất đối xứng:*** HĐH dành ra 1 hoặc 2 bộ xử lý để sử dụng riêng, các bộ xử lý còn lại dùng để điều khiển các chương trình của người sử dụng

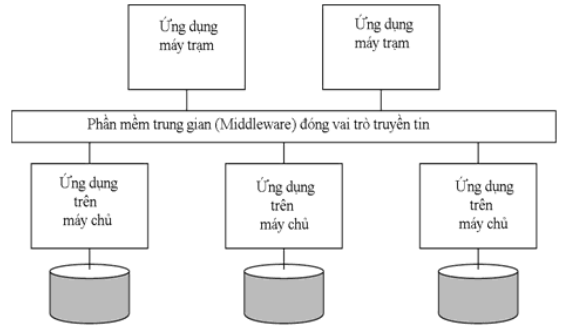
**1.4. Phân loại hệ thống phân tán: 3 loại**

***1.4.1. Hệ thống điện toán phân toán***

| ***Điện toán cụm*** | ***Điện toán lưới*** |
| --- | --- |
| - Các máy tính cá nhân được cài đặt 1 loại HĐH và kết nối với nhau trong mạng tốc độ cao  - Kỹ thuật xử lý song song  - Mỗi cụm bao gồm nhiều nút trong đó có một nút chủ đảm nhiệm chức năng xắp đặt vị trí của các thành viên khác | - Các máy tính liên kết với nhau để giải quyết một vấn đề chung bằng cách chia thành các đơn vị nhỏ hơn  - Không đòi hỏi tính đồng nhất của các nút (có thể khác phần cứng và HĐH)  - Xây dựng trên nguyên tắc đối xứng |

***1.4.2. Hệ thống thông tin phân tán***

* Phải có sự phối hợp xử lý giữa các máy chủ để đảm bảo tính nhất quán

******

***1.4.3 Hệ thống lan tỏa phân tán***

* ***Đặc điểm cơ bản:***
* Thông lượng không ổn định (tính không ổn định về vị trí cũng như tốc độ truyền dẫn)

***1.4.4. Điện toán đám mây*** (tương đồng với điện toán lưới)

* Không cho phép người dùng truy nhập trực tiếp vào tài nguyên, việc truy nhập được thực hiện thông qua các dịch vụ trên nền tảng Internet
* Điện toán lưới ảo hóa tài nguyên, Điện toán đám mây không cho phép người dùng truy nhập trực tiếp vào tài nguyên

**1.5. Mục tiêu hệ thống phân tán**

1. *Thân thiện với người dùng*
2. *Hiệu năng*
3. *Linh hoạt*
4. *Nhất quán*
5. *Chịu lỗi*
6. *An toàn và bảo mật thông tin*

***1.5.2. Tính trong suốt***

Che giấu sự thật các tiến trình và tài nguyên phân tán trên nhiều máy tính, 1 hệ thống phân tán có thể tự trình diễn cho người dùng và các ứng dụng như thể đang chạy trên 1 máy tính

Phân loại trong suốt

* ***Truy nhập:*** che giấu việc thể hiện dữ liệu và cách truy nhập tài nguyên. Đạt được sự đồng thuận về cách thể hiện trên các nền tảng khác nhau
* ***Vị trí:*** Che giấu nơi đặt tài nguyên. Tài nguyên hệ thống có thể được nhân bản nhưng cần đảm bảo tính nhất quán. việc đặt tên đóng vai trò rất quan trọng để đạt được sự trong suốt về vị trí
* ***Di trú:*** Che giấu việc tài nguyên có thể di chuyển sang vị trí khác, khi đặt lại vị trí tài nguyên không làm gián đoạn hoạt động hệ thống. Di chuyển tài nguyên không ảnh hưởng đến việc cung cấp dịch vụ
* ***Nhân bản:*** Che giấu nhân bản tài nguyên. Làm giảm tính nhất quán hệ thống. Giải quyết sự tồn tại nhiều bản sao của tài nguyên
* ***Tương tranh:*** chia sẻ tài nguyên theo cách cộng tác. Cung cấp cái nhìn nhất quán về tài nguyên
* ***Lỗi:*** Che giấu lỗi và vấn đề phục hồi sau khi lỗi xảy ra

**1.5.3. Tính mở**

* Khả năng cung cấp các dịch vụ theo qui tắc chuẩn mô tả cú pháp và ngữ nghĩa của các dịch vụ
* Khả năng dễ dàng chuyển đổi đặc trưng cho mức độ mà người phát triển trên các hệ thống khác nhau có thể tùy biến cùng 1 dịch vụ đã cung cấp
* Khả năng dễ dàng cấu hình hệ thống từ các thành phần khác nhau, có thể dễ dàng thêm thành phần mới hoặc thực hiện những thao tác sửa đổi nhưng không ảnh hưởng đến các thành phần khác
* Các dịch vụ được mô tả bằng ngôn ngữ ***định nghĩa giao diện***
* Các đặc tả giao diện phải: ***đầy đủ*** và ***trung lập***

**1.5.4. Qui mô hệ thống**

* ***3 tiêu chí đo:***
* Có thể mở rộng hệ thống nhưng vẫn dựa trên hạ tầng hiện có (*quan trọng nhất*)
* Có thể mở rộng theo phạm vi địa lý
* Đảm bảo khả năng quản trị hệ thống
* ***Giải thuật KHÔNG TẬP TRUNG có đặc điểm***
* Không máy tính nào có đầy đủ thông tin về trạng thái hệ thống
* Các máy tính ra quyết định chỉ dựa trên thông tin cục bộ
* Lỗi xảy ra trên 1 máy sẽ không ảnh hưởng đến các máy khác và không làm hỏng giải thuật
* Không tồn tại đồng hồ chung cho toàn bộ hệ thống
* ***Khó mở rộng quy mô địa lý do:*** dựa trên phương pháp truyền thông đồng bộ (tiến trình máy khách bị phong tỏa trong thời gian chờ kết quả trả về từ tiến trình máy chủ)
* ***Kỹ thuật xử lý quy mô hệ thống:***
* Che giấu trễ truyền thông (áp dụng truyền thông không đồng bộ)
* Phân tán
* Nhân bản (làm hiệu năng tốt hơn, giảm trễ truyền thông nhưng giảm nhất quán)

**1.6. Kiến trúc hệ thống phân tán**

* ***Mục tiêu thiết kế hệ thống phân tán:*** tách biệt phần ứng dụng với nền tảng bằng cách cung cấp thêm tầng trung gian

***1.6.1. Các mô hình kiến trúc***

| ***Phân tầng***    - Hiệu năng giảm  - Tính chính xác cao  - Khả năng nâng cấp, bảo trì, mở rộng tính năng không ảnh hưởng đến tầng khác | ***Đối tượng phân tán***    - Gọi thủ tục từ xa  - Triển khai dịch vụ khách / chủ  - Thời gian trễ thấp => phù hợp ứng dụng thời gian thực  - Không biết đối tượng gọi có đáp ứng yêu cầu dịch vụ ko, quá tảu  - Phong tỏa bên gọi |
| --- | --- |
| ***Kênh sự kiện***    - Truyền thông giữa các thành phần thông qua việc lan tỏa sự kiện và đăng ký nhận sự kiện  - Phù hợp hệ thống có khả năng phân tải và ***dự phòng nóng*** | ***Dữ liệu tập trung***    Email |

***1.6.2. Kiến trúc hệ thống***

| ***Kiến trúc tập trung*** | ***Kiến trúc không tập trung*** |
| --- | --- |
| - Kiến trúc dọc  - Theo kiến trúc phân tầng  - VD: mô hình khách chủ  - Tầng trung gian đóng vai trò kết nối tiến trình máy khách với tiến trình máy chủ, nó giao tiếp với các tiến trình qua giao diện lập trình ứng dụng  - 3 thành phần:   * Tương tác người dùng * Truy nhập dữ liệu * Xử lý nghiệp vụ | - Kiến trúc ngang hàng  - Dự phòng nóng, cân bằng tải  - Kiến trúc ngang hàng nảy sinh vấn đề làm thế nào có thể tổ chức các tiến trình thành mạng bao trùm |

**CHƯƠNG 2: TRAO ĐỔI THÔNG TIN TRONG HTPT**

* ***Bản chất trao đổi thông tin:*** sử dụng các giao thức truyền tin truyền thống đã quy định trong các tầng của mạng

**2.1. Cơ sở truyền thông**

***2.1.1. Giao thức mạng***

* **Mô hình OSI (7 tầng):**
* ***Ứng dụng:*** cung cấp giao diện phục vụ cho người sử dụng và các ứng dụng khác
* ***Trình diễn:*** mã hóa / giải mã, nén / giải nén và bảo mật dữ liệu
* ***Phiên:*** tạo các phiên làm việc
* ***Giao vận:*** tạo liên kết đầu cuối, điều khiển tốc độ truyền dữ liệu, xử lý lỗi truyền tin
* ***Mạng:*** quản lý địa chỉ logic của các đối tượng tham gia vào mạng, tìm đường đi tốt nhất cho gói tin
* ***Liên kết dữ liệu:*** tạo liên kết giữa 2 thiết bị vật lý kề nhau
* ***Vật lý:*** biến đổi các bit dữ liệu thành các tín hiệu phù hợp với môi trường truyền dẫn và thực hiện thu phát các tín hiệu đó
* ***Giao thức mức thấp:*** thuộc 3 tầng vật lý, liên kết dữ liệu, mạng
* **IP:** giao thức ở tầng mạng, loại không liên kết
* Đảm bảo vận chuyển dữ liệu 1 cách tốt nhất có thể được giữa các thiết bị đầu cuối của người sử dụng
* ***Giao thức tầng vận tải:***
* Quản lý việc trao đổi thông tin giữa các tiến trình trên các thiết bị đầu cuối của người sử dụng
* **TCP:** giao thức liên kết
* **UDP:** giao thức không liên kết
* ***Giao thức mức cao:*** tầng phiên, tầng trình diễn và tầng ứng dụng
* ***Giao thức tầng trung gian:*** tầng ứng dụng

***2.1.2. Phân loại truyền thông***

* ***Truyền thông bền bỉ:*** các thông điệp của người dùng được lưu trữ trong hệ thống cho đến khi chuyển thành công đến người nhận, bên gửi và bên nhận hoạt động hoàn toàn độc lập với nhau
* ***Truyền thông tạm thời (nhất thời):*** chỉ lưu giữ thông điệp trong thời gian gửi và nhận, nghĩa là bên gửi và bên nhận phụ thuộc lẫn nhau
* ***Truyền thông đồng bộ:*** bên gửi sẽ bị phong tỏa cho đến khi biết chắc chắn yêu cầu của mình đã được bên nhận xử lý

vận hành tốt trong mạng cục bộ, nơi mà truyền thông giữa các máy tính chỉ vài µs

* ***Truyền thông không đồng bộ:*** bên gửi tiếp tục thực hiện công việc của mình sau khi đã gửi thông điệp đến hệ thống trung gian

**2.2. Gọi thủ tục từ xa**

* Mô hình khách chủ
* Truyền tham số được thực hiện bằng phương pháp truyền giá trí hay truyền địa chỉ, sử dụng cơ chế con trỏ hoặc tham chiếu
* 10 bước thực hiện:

1. Thủ tục trên máy khách gọi stub như phương pháp gọi thủ tục truyền thống.
2. Stub tạo thông điệp và chuyển đến hệ điều hành của máy khách .
3. Hệ điều hành của máy khách gửi thông điệp đến hệ điều hành của máy chủ.
4. Hệ điều hành của máy chủ chuyển thông điệp đến Skeleton .
5. Skeleton giải mã thông điệp thành các tham số và gọi thủ tục xử lý tương ứng.
6. Máy chủ thực hiện lời gọi thủ tục và trả về giá trị cho Skeleton.
7. Skeleton đóng gói tham số giá trị trả về thành thông điệp và chuyển đến hệ điều hành của máy chủ.
8. Hệ điều hành của máy chủ chuyển thông điệp đến hệ điều hành của máy khách.
9. Hệ điều hành của máy khách nhận thông điệp và chuyển cho Stub.
10. Stub giải mã kết quả và trả về các tham số theo yêu cầu của thủ tục đã gọi.

**2.3. Truyền thông điệp**

***2.3.1. Truyền thông điệp nhất thời***

* ***Các hàm nguyên thủy:*** Mức độ trừu tượng thấp

***2.3.2. Truyền thông điệp bền bỉ***

| **Tên hàm** | **Ý nghĩa** |
| --- | --- |
| **Put** | Thêm thông điệp vào hàng đợi đã xác định |
| **Get** | Phong tỏa cho đến khi hàng đợi không rỗng và xóa thông điệp đầu tiên |
| **Poll** | Kiểm tra thông điệp có trong hàng đợi hay không và xóa thông điệp đầu tiên, không phong tỏa tiến trình gọi |
| **Notify** | Cài đặt trình điều khiển để thông báo khi thông điệp được đặt vào hàng đợi |

**2.4. Truyền thông luồng**

* Đòi hỏi tính đồng bộ giữa các luồng dữ liệu và đồng thời sự liên tục của thông tin
* Sử dụng phương thức đẳng thời
* Gọi một thủ tục từ xa hoặc truyền thông điệp không phụ thuộc vào thời điểm thực hiện

**2.5. Truyền thông theo nhóm**

* Phương pháp truyền thông ngang hàng, bên gửi phải biết thông tin dùng cho việc thiết lập kênh truyền đến tất cả các thành viên nhận
* Nguyên lý lan truyền thông tin

***2.5.1. Truyền thông theo nhóm mức ứng dụng:***

* ***Tổ chức hình cây:*** Tổ chức các nút thành một cây theo giải thuật Chord để tìm kiếm nút chịu trách nhiệm đối với khóa ngẫu nhiên.
* ***Quảng bá trong nhóm:*** giảm thiểu các nút trung gian mà thông điệp không được sử dụng

***2.5.2. Lan truyền ngẫu nhiên***

* Nguyên lý lan truyền dịch, không cần nút điều phối, dùng 1 thông tin cục bộ vẫn truyền thông nhanh chóng trong tập hợp lớn các nút

**CHƯƠNG 3: ĐẶT TÊN TRONG HTPT**

* ***Đặc điểm hệ thống đặt tên:***
* Cài đặt trên nhiều máy tính
* Qui tắc đặt tên phải đáp ứng yêu cầu về hiệu năng khi mở rộng qui mô hệ thống, tạo điều kiên tìm kiếm đối tượng 1 cách tốt nhất

**3.1. Tên, định danh và địa chỉ**

| ***Tên*** | ***Định danh*** | ***Địa chỉ*** |
| --- | --- | --- |
| 1 xâu các bit hoặc ký tự dùng để tham chiếu đến 1 thực thể | - Tham chiếu đến duy nhất 1 thực thể  - Mỗi thực thể chỉ có 1 định danh  - Không được tái sử dụng định danh | - Điểm truy cập đến thực thể  - 1 tên có thể gồm nhiều địa chỉ  - Thực thể có thể thay đổi địa chỉ  - Có thể tham chiếu tới các tài nguyên = tên |

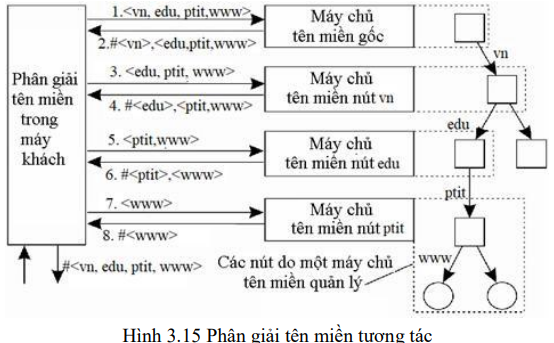
**3.2. Đặt tên và các giải pháp tìm kiếm**

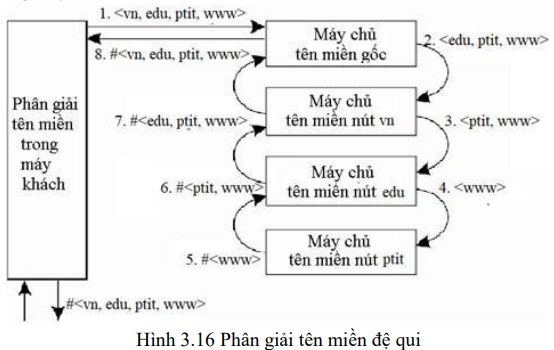
***3.2.1. Đặt tên phi cấu trúc:***

* Tên phẳng, chỉ gồm chuỗi các bit ngẫu nhiên không chứa bất kỳ thông tin liên quan tới thực thể
* ***4 giải pháp xác định điểm truy nhập tới thực thể:***
* Giải pháp đơn giản
* Dựa trên nguồn gốc
* Hàm băm phân tán
* Giải pháp phân cấp

***3.2.2. Đặt tên có cấu trúc***

* Phù hợp các máy tính nhưng nó hoàn toàn không thân thiện với con người, con người cần những tên đơn giản dễ đọc và dễ nhớ
* Phân giải tên miền: từ phải sang trái

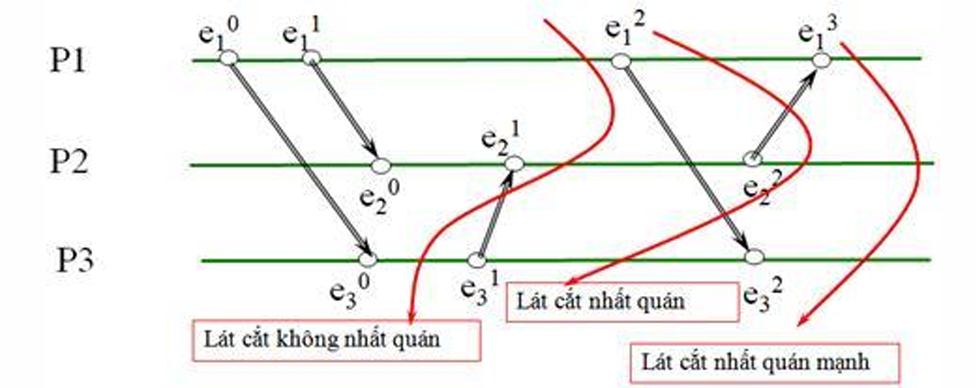




**CHƯƠNG 4: ĐỒNG BỘ VÀ CÁC GIẢI THUẬT PHÂN TÁN**

**4.1. Trạng thái toàn cục:**

* Tại một thời điểm bất kỳ, trạng thái toàn cục của một hệ thống phân tán được đặc trưng bởi trạng thái của từng tiến trình và các thông điệp đang lưu chuyển trong hệ thống.
* Sử dụng lát cắt để xác định trạng thái toàn cục



* **Lát cắt nhất quán:** với tất cả các nút e trong phần được cắt, nếu tồn tại f -> e thì f cũng phải thuộc phần được cắt
* **Lát cắt nhất quán mạnh:** tất cả quan hệ nhân quả cùng phải thuộc lát cắt
* Còn lại là **lát cắt không nhất quán**

**CHƯƠNG 5: TIẾN TRÌNH TRONG CÁC HỆ THỐNG PHÂN TÁN**

* Tiến trình: Một chương trình đang chạy
* Nhiệm vụ chính của hệ điều hành: quản lý và lập lịch cho các tiến trình

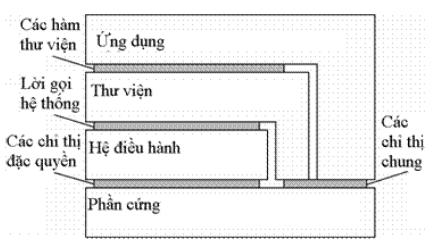
**5.1. Các luồng**

* ***Khái niệm:***
* Mỗi chương trình khi chạy cần cung cấp 1 bộ xử lý => để đáp ứng yêu cầu cùng 1 lúc chạy được nhiều chương tình thì HĐH tạo ra bộ ***xử lý ảo*** cho mỗi chương trình
* Tiến trình như là 1 chương trình đang chạy trên bộ xử lý ảo
* Luồng thực hiện đoạn mã chương trình độc lập với luồng khác
* Luồng không cố đạt độ trong suốt cao nếu thao tác đó làm suy giảm hiệu năng hoạt động
* ***Cài đặt luồng***
* Có hai cách tiếp cận cơ bản: tạo thư viện luồng mức người dùng hoặc mức lõi.
* Kết hợp luồng ở mức người dùng và mức lõi, những luồng như vậy gọi là tiến trình con LWP
* Mỗi tiến trình con sẽ tạo ra một ngăn xếp riêng chứa các luồng sẽ được thực hiện, các luồng được quản lý trong bảng luồng và được bảo vệ bằng cơ chế biến đồng bộ.
* ***Luồng trong hệ thống phân tán:***

| ***Mô hình*** | ***Đặc điểm*** |
| --- | --- |
| Đa luồng | Song song, phong tỏa hệ thống |
| Đơn luồng | Tuần tự, phong tỏa hệ thống |
| Máy trạng thái | Song song, không phong tỏa hệ thống |

**5.2. Ảo hóa**

* ***Xuất phát từ nhu cầu:*** tài nguyên vật lý chỉ có 1 nhưng nhiều thành phần cùng sử dụng chung tài nguyên và muốn tài nguyên thuộc về mình => tương tranh giữa các tiến trình
* ***Kiến trúc máy ảo:*** 4 loại giao diện



**5.3. Máy khách:**cung cấp giao diện cho người sử dụng tương tác với máy chủ, tiền xử lý dữ liệu trước khi chuyển yêu cầu đến máy chủ

**5.4. Máy chủ:**chạy các tiến trình cung cấp dịch vụ theo yêu cầu của máy khách, thường là những máy tính có cấu hình đủ lớn để luôn sẵn sàng đáp ứng yêu cầu dịch vụ máy khách

**5.5. Di trú mã**

* ***Khái niệm:*** Di chuyển các chương trình hoặc đoạn mã chương trình để tăng hiệu suất và độ linh hoạt hệ thống
* ***Tư tưởng cơ bản:*** dựa trên hiệu năng tổng thể, chuyển tiến trình từ máy tính đang có tải cao sang máy có tải thấp
* ***Di trú mã trong hệ thống không đồng nhất***

1. Chuyển các trang bộ nhớ sang máy mới và sẽ cập nhật lại những trang bị thay đổi trong quá trình di trú.

2. Ngừng máy ảo, di trú bộ nhớ và khởi tạo máy ảo mới.

3. Khởi tạo các tiến trình trên máy ảo mới và sao chép các trang bộ nhớ theo yêu cầu của tiến trình đó.

**CHƯƠNG 6: QUẢN TRỊ GIAO TÁC VÀ ĐIỀU KHIỂN TƯƠNG TRANH**

* Nếu nhiều tiến trình đồng thời truy xuất dữ liệu thì có thể dẫn tới hiện tượng tương tranh
* Hậu quả tất yếu sẽ làm suy giảm hiệu năng hệ thống và thậm chí có thể dẫn

đến sai lệch thông tin

* ***Điều khiển tương tranh:*** cơ chế cho phép nhiều giao tác thực hiện đồng thời mà không xảy ra sự tranh chấp giữa các giao tác

**6.1. Khái niệm giao tác**

* Một hoặc nhiều câu lệnh truy xuất dữ liệu và được thực hiện như 1 đơn vị thống nhất
* Chia thành 3 loại: phẳng, lồng ghép, phân tán
* ***4 tính chất:***
* Nguyên tử (Atomic)
* Nhất quán (Consistent)
* Cô lập (Isolated)
* Bền vững (Durable)
* ***Hoạt động trong giao tác:***
* Mở giao tác
* Thao tác với dữ liệu
* Đóng giao tác bằng lệnh cam kết (commit) hoặc hủy bỏ (abort, rollback)

**6.2. Các phương pháp điều khiển tương tranh**

* ***Điều khiển tương tranh bi quan:*** nhìn nhận hệ thống luôn tiềm ẩn tương tranh
* Để truy xuất một mục dữ liệu, giao tác Ti trước tiên phải khóa mục dữ liệu này và khi thực hiện xong thì phải giải phóng khóa
* Tư tưởng: các thao tác trên một đơn vị dữ liệu nếu có xung đột thì tại một thời điểm chỉ cho phép một giao tác thực hiện (dựa trên cơ chế khóa)
* Deadlock là trạng thái mỗi thành viên của nhóm các giao tác đang chờ nhau giải phóng khóa, phòng ngừa:
* Khi bắt đầu giao tác sử dụng khóa tất cả các mục dữ liệu.
* Mỗi giao tác yêu cầu khóa trên các mục dữ liệu theo thứ tự đã định nghĩa trước
* Mỗi khóa có khoảng thời gian giới hạn, sau thời gian đó sẽ không được bảo vệ.
* ***Điều khiển tương tranh lạc quan:*** nhìn nhận không xảy ra tương tranh và khi nào xảy ra thì mới giải quyết.
* Cho phép các giao tác được phép tiếp tục nếu không có xung đột với các giao tác khác, nếu phát hiện xung đột thì sẽ hủy bỏ giao tác nào đó.
* Mỗi giao tác gồm ba pha:
* Pha đọc: sử dụng bản tạm thời cho mỗi mục dữ liệu được cập nhật
* Pha phê chuẩn: Kiểm tra xem có xung đột hay không
* Pha ghi: Nếu được phê chuẩn không có xung đột thì chuyển bản dữ liệu tạm thời thành vĩnh viễn
* ***Sử dụng khóa:*** là một cách tiếp cận điển hình về điều khiển tương tranh bi quan, nhược điểm cơ bản của nó là tăng tải xử lý cho hệ thống
* ***Điều khiển tương tranh dựa trên nhãn thời gian:*** Lamport

**CHƯƠNG 7: PHỤC HỒI VÀ TÍNH CHỊU LỖI**

**7.1. Tính chịu lỗi:**

* **Tiêu chí:**
* Khả năng sẵn sàng phục vụ hệ thống
* Độ tin cậy của hệ thống
* Độ an toàn của hệ thống
* Khả năng bảo trì hệ thống
* Phân loại:
* Lỗi nhất thời
* Lỗi lặp
* Lỗi lâu dài
* Máy chủ lỗi nghiêm trọng
* Máy chủ xử lý lỗi
* Lỗi thời gian
* Lỗi kết quả xử lý
* Lỗi không xác định

**7.2. Các biện pháp đảm bảo chịu lỗi**

* Che dấu bằng biện pháp dư thừa:
* **Dư thừa thông tin:** bổ sung thêm các bit dư thừa để phát hiện lỗi và phục hồi lỗi
* **Dư thừa thời gian:** khi một hoạt động đã được thực hiện, nếu dư thừa thời gian nó có thể được thực hiện lại, phù hợp khi lỗi là **nhất thời và lỗi chập chờn**
* **Dư thừa vật lý:** bổ sung thêm tài nguyên vật lý

**7.3. Các dạng commit: 2 pha, 3 pha,...**

**7.4. Phục hồi:**

* Phục hồi lùi: chuyển trạng thái của hệ thống về trạng thái gần nhất trước khi xảy ra lỗi và cho hệ thống chạy lại từ điểm đó

-> Phải có **CHECK\_POINT**

* Phục hồi tiến: chuyển trạng thái của hệ thống đang từ lỗi thành trạng thái khác để tiếp tục thực hiện
* Đánh dấu kiểm tra phối hợp:
* Tiến trình điều phối gửi CHECKPOINT\_REQUEST
* Nhận được CHECKPOINT\_REQUEST, tất cả thành viên ngừng gửi, thống điệp nhận được đưa vào hàng đợi
* Điều phối gửi CHECKPOINT\_DONE
* Nhận được CHECKPOINT\_DONE các tiến trình khác tiếp tục
* **Tiến trình mồ côi:** Sống sót sau sự sụp đổ của tiến trình khác nhưng trạng thái không nhất quán với tiến trình sụp đổ đã được phục hồi

**CHƯƠNG 8: NHẤT QUÁN VÀ NHÂN BẢN**

**8.1. Mô hình nhất quán lấy dữ liệu làm trung tâm:**

1. **Nhất quán liên tục**
2. **Nhất quán theo thứ tự giao tác**
   1. **Nhất quán tuần tự:** Các thao tác đọc/ghi dữ liệu phải tuân thủ thứ tự đã được đặt trước
   2. **Nhất quán nhân quả:**

* Phân biệt có nhân quả hay không
* Nếu a->b thì mọi cái nhìn thấy a trước
* Nếu a->b nhân quả tiềm năng thì mọi thực thể phải nhận biết giống hệt cùng 1 thứ tự
  1. **Nhất quán hàng đợi:**
  2. **Nhất quán nghiêm ngặt:** Thao tác đọc phải trả về kết quả của thao tác ghi gần nhất trên mục dữ liệu đó, sử dụng **thời gian tuyệt đối** -> khó với hệ thống phân tán

1. **Nhất quán theo nhóm giao tác**
2. **Nhất quán yếu:** Quan tâm đến trật tự các nhóm lệnh bằng việc sử dụng các **biến đống bộ hoá**. Tức là sẽ có 1 nhóm lệnh, trong nhóm lệnh đó thì thứ tự thế nào cũng được, những thứ tự các nhóm thì phải giữ nguyên.
3. **Nhất quán đi ra**
4. **Nhất quán bảng dữ liệu:** Có 2 lệnh: **(chỉ thực hiện dữ liệu dùng chung)**

* acq: lệnh chiếm giữ để báo muốn vào vùng tới hạn
* rel: lệnh giải phóng để rời khỏi vùng tới hạn

1. **Nhất quán mục dữ liệu:** Giống bảng dữ liệu, tiến trình nào muốn sử dụng mục dữ liệu phải đợi cho tiến trình khác giải phóng mục dữ liệu đó
2. **Nhất quán và gắn kết**

**8.2. Mô hình nhất quán lấy máy khách làm trung tâm:**

1. **Nhất quán sau cùng:** Sau mỗi lần cập nhật thì tất cả bản sao phải giống nhau -> cập nhật bản sao ngay sau khi cập nhật bản chính

Nếu nhiều quá thì cho vào hàng đợi xử lý dần, sử dụng bất đông bộ

1. **Nhất quán đọc đều:** Đọc cho ra kết quả gần nhất cập nhật, đọc không bao giờ cũ hơn bản sao đọc trước đó
2. **Nhất quán ghi đều:** Ghi trên cùng 1 tiến trình phải theo trình tự
3. **Nhất quán đọc kết quả ghi:** Luôn cho ra kết quả ghi mới nhất
4. **Nhất quán ghi sau khi đọc**