**Đặt vấn đề**

Trong những năm qua, cùng với sự phát triển như vũ bão của khoa học kĩ thuật, ngành công nghệ thông tin cũng biến đổi không ngừng để đáp ứng với mọi nhu cầu mới của con người. Trong số những phát minh vĩ đại nhất của thế kỉ trước, mạng máy tính ( Computer Network) là một hệ thống đem lại nhiều lợi ích to lớn nhất cho nhân loại. Mạng máy tính ngày nay cho phép hàng tỉ máy tính trên trái đất đã và đang kết nối và làm việc việc với nhau với tốc độ cao ( từ vài chục đến hàng nghìn kbps).

Điều kì diệu trên đã trở thành hiện thực và ngày càng phát triển mạnh mẽ hơn nữa chỉ khi có một nhân tố quan trọng làm nền tảng cơ sở lí thuyết cho chúng đã được nghiên cứu. Đó chính là những lí thuyết nghiên cứu liên quan đến hệ phân tán. Nhờ có lí thuyết hệ phân tán, mà các vi xử lí đơn lẻ - nằm trên các máy tính đơn lẻ (kèm theo là các thiết bị ngoại vi và các thiết bị đầu cuối làm việc từ xa)- mà chúng vẫn thường được gọi là các hệ thống tập trung (Centralized Systems) đã được kết nối lại, làm việc trên nhiều giao thức, nhiều kiểu kết nối và trên nhiều lãnh thổ khác nhau. Và khi các hệ thống tập trung đó đã được kết nối lại theo nhiều nhóm, nhiều cách thức khác nhau và làm việc dựa trên những nguyên lí cơ bản của hệ phân tán thì chúng sẽ được gọi là các hệ thống phân tán (Distributed Systems), điển hình nhất của các hệ phân tán chính là mạng máy tính mà chúng đa đang sử dụng ngày nay.

Trong báo cáo này chúng ta sẽ tìm hiểu về vấn đề chia sẻ tài nguyên trong hệ phân tán. Báo cáo sẽ đề cập đến chia sẻ tài nguyên trong kiến trúc và nguyên lý của hệ phân tán.

**Nội dung chính**

1. **Tổng quan về hệ phân tán :**

Trước khi đi vào tìm hiểu các nguyên lí về hệ phân tán, chúng ta cần tìm hiểu tổng quan về hệ phân tán.

Có nhiều định nghĩa khác nhau về hệ phân tán, dưới đây là ba định nghĩa được nhiều người chấp nhận nhất:

Định nghĩa 1: Hệ phân tán là một hệ thống có chức năng và dữ liệu phân tán trên các trạm (máy tính) được kết nối với nhau qua một mạng máy tính.

Định nghĩa 2: Hệ phân tán là một tập các máy tính tự trị được kết nối với nhau bởi một mạng máy tính và được cài đặt phần mềm hệ phân tán.

Như vậy:

Hệ phân tán = Mạng máy tính (+) Phần mềm hệ phân tán.

( Distributed System = Computer Network (+) Distributed System Software)

Định nghĩa 3: Hệ phân tán là một tập các máy tính độc lập giao tiếp với người sử dụng như một hệ thống thống nhất và toàn vẹn ( A single coherent system).

Về cơ bản, định nghĩa 3 cũng tương đương với định nghĩa 2. Bởi vì một tập hợp các máy tính độc lập muốn trở thành một hệ thống thống nhất và toàn vẹn thì nó phải được kết nối thành một mạng máy tính, trên đó cài đặt phần mềm hệ phân tán ( chẳng hạn hệ điều hành phân tán – Distributed Operating System)

Bản thân một mạng máy tính cũng là một hệ phân tán. Bởi vì trên đó có các phần mềm cài đặt các giao thức truyền thông là các phần mềm hệ phân tán.

Một số ví dụ khác về các hệ phân tán:

Các hệ điều hành phân tán : Dòng hệ điều hành phân tán xuất phát từ Unix là Unix Berkeley Software Distribution ( hay Unix BSD). Network File System (NFS) của Sun phát triển dựa vào dòng Unix BSD. Ngoài ra có các hệ điều hành phân tán như Amoeba, Chorus, Mach,…dành cho các công ti, các viện, trung tâm nghiên cứu.

Các dịch vụ thông tin trên mạng như Email, E-commerce, E-learning, E-healt,…

Hệ thống đặt vé máy bay: Ở các nước phát triển,hệ thống này đã được đưa vào sử dụng từ lâu. Còn ở Việt Nam mới chỉ đang trong giai đoạn thử nghiệm. Chẳng hạn cách đây vài tháng VietNam AirLine đã thử nghiệm hình thức đặt vé máy bay qua mạng internet, khuyến mại giảm giá vé, kết quả rất khả quan.

**Mục tiêu của hệ phân tán**: Mục tiêu chính của hệ phân tán là để giải quyết bài toán chia sẻ tài nguyên ( Resource Sharing). Nội dung bài toán đó như sau:

*Có một tập hữu hạn các tài nguyên ( gồm các máy tính, đường truyền, thiết bị mạng, phần mềm, dữ liệu trên các máy). Và có một tập hữu hạn những người sử dụng ở các vị trí khác nhau và có thể gia tăng nhanh về số lượng người.*

*Vấn đề đặt ra: Các giải pháp chia sẻ tối ưu những tài nguyên trên giữa những người sử dụng là gì?*

1. **Khái niệm phần cứng trong hệ phân tán :**
2. **Phân loại máy tính.**

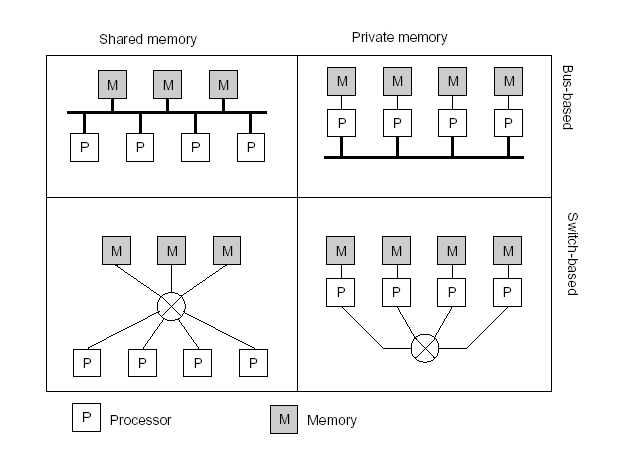
Có hai loại máy tính:

* Các loại máy tính có chia sẻ bộ nhớ (Shared memory): các loại máy đa xử lý (multiproccessor).
* Các máy tính không chia sẻ bộ nhớ (Private memory): các hệ thống multicomputors

Trong mỗi loại lại chia tiếp theo mạng kết nối bus – based chỉ có một

đường kết nối và switch – base có nhiều dùng kết nối từ máy này sang

máy khác



Hình 1: Hai cách tổ chức vi xử lý và bộ nhớ trong hệ máy tính phân tán.

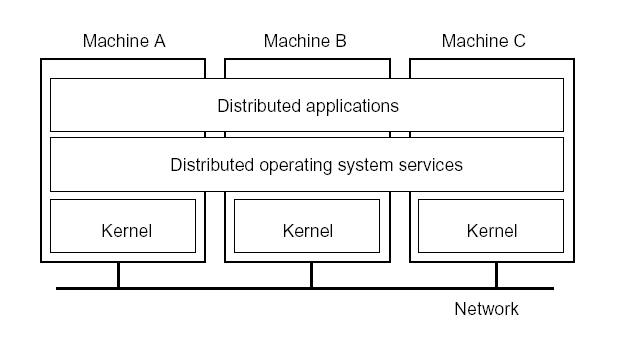
1. **Hệ thuần nhất / hệ không thuần nhất :**

Hệ thống thuần nhất: mạng máy tính cùng sử dụng một công nghệ, các bộ xử lý là như nhau, truy cập đến cùng một bộ nhớ giống nhau. Thường dùng trong hệ thống có tính toán song song.

Hệ không thuần nhất: những máy tính khác nhau kết nối với nhau.

1. **Khái niệm phần mềm trong hệ phân tán :**
2. DOS ( distributed operating system ) :

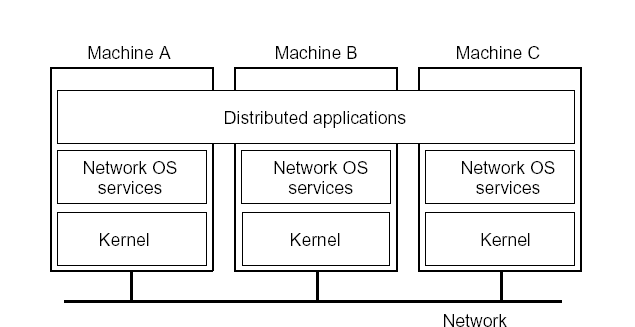
* Là hệ điều hành cho các hệ multiproccessor và các hệ homogenous multicomputer.
* Mục tiêu là ẩn giấu và cung cấp các dịch vụ quản trị tài nguyên.
* Đặc điểm là các dịch vụ có thể được thực hiện bởi các lời triệu gọi từ xa.



Hình 2. Cấu trúc chung của DOS

1. NOS ( network operaring system) :

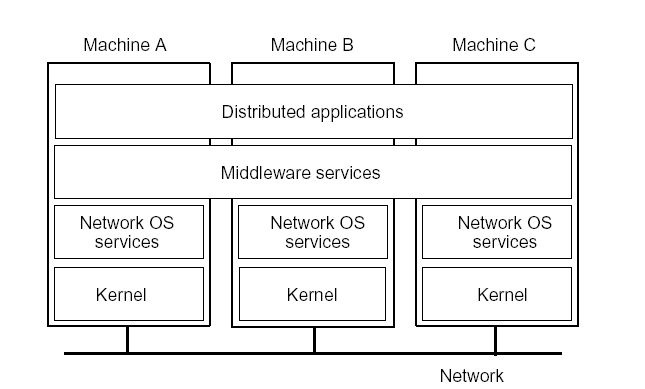
* Là hệ điều hành cho các hệ thống heterogenous multicomputer (LAN, WAN).
* Mục tiêu của NOS là cung cấp các dịch vụ từ xa.



Hình 3 . Cấu trúc chung của NOS

1. Middleware :

Là tầng phụ nằm giữa tầng dịch vụ của NOS và tầng ứng dụng phân tán.



Hình 4. Cấu trúc chung của một hệ middleware

1. **Đồng bộ những tiến trình khi truy cập cùng một lúc một tài nguyên**

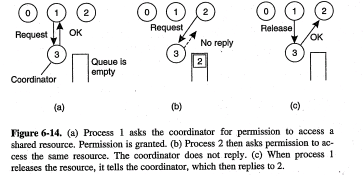
Nguyên tắc cơ bản của hệ phân tán là sự đồng thời và cộng tác của những đa tiến trình. Trong nhiều trường hợp , điều này có nghĩa là những tiến trình cần truy cập cùng một lúc cùng một tài nguyên. Để tránh điều này giải pháp là cho phép xử lý truy cập theo kiểu lọai trừ lẫn nhau.

Thuật toán phân tán loại trừ lẫn nhau có thể được phân lớp thành 2 giải pháp khác nhau:

* 1. Phương pháp thứ nhất đạt được bằng cách đưa ra một thông điệp đặc biệt giữa các tiền trình ,được gọi là biểu tượng – token. Chỉ có 1 token sẵn sàng và ai là người có token này được phép truy cập đến tài nguyên chia sẻ.Phương pháp này có một số đặc tính quan trọng
* Đầu tiên là nó phụ thuộc vào cách làm thế nào để tổ chức các tiến trình ,những đặc tính này dễ dàng bảo đảm cho mọi tiến trình có một sự thay đổi trong truy cập tài nguyên. Nói cách khác ,những đặc tính này tránh sự thiếu tài nguyên.
* Do **deadlock** ,tiến trình phải đợi những tiến trình khác để được xử lý, có thể tránh deadlock một cách dễ dàng bằng cách đóng góp vào sự đơn giản của các tiến trình. Không may là trở ngại chính của token-base solution lại nghiêm trọng hơn, một chương trình con phức tạp được phân phối cần được chạy để chắc chắn rằng một token mới đựoc tạo ra nhưng cái chính là token đó phải token duy nhất .
  1. Phương pháp thứ hai : một tiến trình muốn truy cập vào tài nguyên thì phải được sự cho phép của tiến trình khác.Có các cách để được sự cho phép và được chỉ ra dưới đây :

1. Thuật toán tập trung :

* Cách dễ hiểu nhất để đạt được sự loại trừ lẫn nhau trong hệ thống phân tán là giả lập cách nó thực hiện trong hệ thống một Bộ xử lý. Một tiến trinh được bầu như một bộ điều phối. Bất cứ lúc nào một tiến trình muốn truy cập những tài nguyên chia sẻ, nó gửi một thông điệp yêu cầu tới bộ điều phối đang thống kê xem loại tài nguyên nào mà tiến trình muốn truy cập và xin phép truy cập. Nếu như không có tiến trình nào đang truy cập tào nguyên, bộ điều phối sẽ gửi lại tiến trình xin phép một thông điệp cho phép truy cập hệ thống.



1. Thuật toán không tập trung :

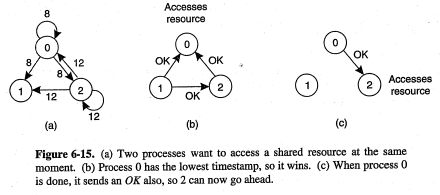
* Giải pháp là mở rộng những bộ phân phối tập trung theo cách sau: Mỗi tài nguyên được gán một bản sao **n lần.** Mỗi bản sao có bộ phân phối của nó để điều khiển việc truy nhập bởi những tiến trình thực thi đồng thời.
* Dù vậy, mỗi khi một tiến trình muốn truy cập tài nguyên nó phải được sự cho phép của m >= n/2 bộ điều phối. Không giống như giải pháp tập trung được đưa ra ở trên(trong trường hợp b của ví dụ), khi một bộ điều phối không đưa ra sự đồng ý để truy cập tài nguyên, nó sẽ cho tiến trình yêu cầu tài nguyên biết. Khi một bộ điều phối bị hỏng, nó nhanh chóng được khôi phục nhưng lại không nhớ được số vote mà nó đã có trước đó., hay nói cách khác là bộ điều phối tự khởi động lại ở thời điểm bất kỳ mà lỗi xảy ra. Nếu như yêu cầu truy cập tài nguyên bị từ chối thì nó sẽ được trả lại một biến thời gian chờ chọn ngẫu nhiên và cố thực hiện lần sau.

1. Thuật toán phân tán :

* Khi một tiến trình muốn truy cập vào một tài nguyên chia sẻ, nó tạo một thông điệp bao gồm tên của tài nguyên, số xử lý của nó và thời gian(theo logic) hiện tại. Sau đó nó gửi thông điệp này tới các tiến trình khác và chính nó. Việc gửi các thông điệp đi là đáng tin cậy và không có thông điệp nào bị mất.
* Khi tiến trình nhận một thông điệp yêu cầu từ tiến trình khác, ứng xử của nó phụ thuộc vào trạng thái của nó với tài nguyên được đặt tên trong thông điệp. Có 3 trường hợp khác nhau được phân biệt rõ ràng:

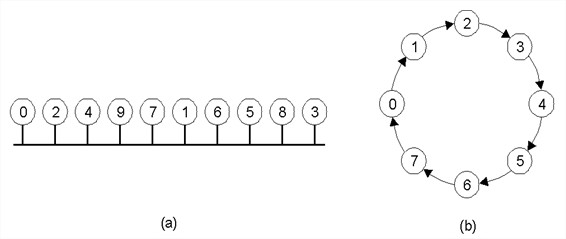
1. Nếu bên nhận đang không hoặc không muốn truy nhập vào tài nguyên, nó sẽ gửi lại thông điệp là OK tới bên gửi.
2. Nếu bên nhận vừa truy cập tài nguyên, nó đơn giản là không phản hồi lại thông điệp yêu cầu, thay vào đó, nó xếp hàng thông điệp yêu cầu đó.
3. Nếu bên nhận cũng muốn truy cập tài nguyên nhưng chưa được phép ,nó sẽ so sánh nhãn thời gian ( timestamp ) của thông điệp gửi đến với timestamp chứa trong thông điệp mà nó gửi đi cho những tiến trình khác. Nếu thông điệp đến có timestamp thấp hơn,bên nhận sẽ gửi thông điệp OK,nếu không thì nó không gửi gì cả.

* Sau khi gửi các gói tin yêu cầu cho phép, một tiến trình đợi đến khi các tiến trình khác cho phép và ngay sau khi các tiến trình cho phép, tiến trình này truy cập tài nguyên. Khi nó kết thúc, nó gửi 1 thông điệp OK đến tất cả các tiến trình khác ở trong hàng đợi của nó và xóa nội dung hàng đợi đó.



1. Giải thuật vòng với thẻ bài ( Tokenring Algorithm)

* Giả thiết tất cả các tiến trình được sắp xếp trên một vòng tròn logic, các tiến trình đều được đánh số và đều biết đến các tiến trình cạnh nó.



Ví dụ theo giải thuật vòng thẻ bài

* Bắt đầu quá trình truyền, tiến trình 0 sẽ được trao một thẻ bài. Thẻ bài này có thể lưu hành xung quanh vòng tròn logic. Nó được chuyển từ tiến trình k đến tiến trình (k+1) bằng cách truyền thông điệp điểm – điểm. Khi một tiến trình giành được thể bài từ tiến trình bên cạnh nó sẽ kiểm tra xem có thể vào vùng tới hạn hay không. Nếu không có tiến trình khác trong vùng tới hạn nó sẽ vào vùng tới hạn.
* Sau khi hoàn thành phần việc của mình nó sẽ nhả thẻ bài ra, thẻ bài có thể di chuyển tự do trong vòng tròn. Nếu 1 tiến trình muốn vào vùng tới hạn thì nó sẽ giữ lấy thẻ bài, nếu không nó sẽ để cho thẻ bài truyền qua.

1. **Mô hình nhất quán :**

* Mô hình nhất quán được sử dụng rộng rãi nhất là mô hình lấy dữ liệu làm trung tâm. Trong mô hình này, bất kì người sử dụng nào truy cập vào kho dữ liệu cũng sẽ nhìn thấy các thao tác được sắp xếp theo mô hình. Điều này trái ngược với các mô hình lấy client làm trung tâm nơi client được yêu cầu một mô hình nhất quán cụ thể và các client khác nhau sẽ nhìn thấy các thao tác theo những trật tự khác nhau.
* Kho dữ liệu có thể được đọc hay ghi bởi bất cứ tiến trình nào trong một hệ phân tán. Tuy nhiên dữ liệu ghi vào một bản sao cục bộ phải đảm bảo cũng được truyền tới tất cả các bản sao ở xa. Vì nguyên nhân này mà các mô hình nhất quán đã ra đời, mục đích là để giúp chúng ta có thể hiểu rõ về các cơ chế khác nhau được sử dụng để thực hiện phần việc này.
* Một mô hình nhất quán có thể được coi là một bản hợp đồng giữa một kho dữ liệu của hệ phân tán với các các tiến trình của nó. Nếu các tiến trình đồng ý với các điều khoản của hợp đồng thì kho dữ liệu sẽ hoạt động đúng như tiến trình mong muốn.

