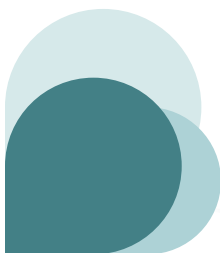




# Phát triển phần mềm phân tán



GV. TS. Vũ Thị Hương Giang  
Viện CNTT&TT



## CHƯƠNG I. KHÁI NIỆM CHUNG

1. Tổng quan về hệ phân tán, ứng dụng phân tán và phần mềm phân tán
2. Phát triển phần mềm phân tán : mục đích, cách tiếp cận và các vấn đề cần giải quyết
3. Vai trò của phần dẻo, truyền thông và hệ điều hành cũng như các dịch vụ hỗ trợ khác trong quá trình phát triển phần mềm phân tán, ví dụ minh họa



# 1. HỆ PHÂN TÁN, ỨNG DỤNG PHÂN TÁN VÀ PHẦN MỀM PHÂN TÁN



## 1.1. Hệ phân tán (distributed system)

[Tanenbaum]:

“A collection of independent computers that appears to its users as a single coherent system.”

3 đặc tính chính:

- Tính tự trị (autonomous) của máy tính
- Phần mềm chỉ cho người dùng thấy một hệ thống duy nhất
- Dễ dàng mở rộng hệ thống mà không cần báo trước cho người dùng

## I.2. Tính toán phân tán (distributed computing)

- [Webopedia] :  
"A type of computing in which different components and objects comprising an application can be located on different computers connected to a network."
- [Wikipedia] :  
"Decentralized and parallel computing, using two or more computers communicating over a network to accomplish a common objective or task."

## I.3. Khác biệt giữa hệ thống (system), phần mềm (software) và ứng dụng (application)

- Hệ thống:
  - Quản lý tài nguyên chung, cơ sở hạ tầng, cung cấp các chức năng quản lý chung qua các giao diện có mức độ trừu tượng cao
  - Gắn liền với các thiết bị liên quan, che giấu sự phức tạp và việc trao đổi thông tin giữa các thiết bị này
  - Ví dụ: hệ điều hành, hệ thống truyền thông
- Phần mềm
  - Chương trình máy tính + tài liệu liên quan (thường tham chiếu đến khái niệm dữ liệu của máy tính)
  - Không nhất thiết phải ở trạng thái thực thi (executable) hay gắn liền với một hệ điều hành cụ thể
  - Không nhất thiết phải tương tác với người dùng trong quá trình hoạt động
- Ứng dụng
  - Một kiểu phần mềm đặc biệt, nhằm mục đích thực hiện một nhiệm vụ nhất định
  - Chỉ thực thi được trên hệ điều hành cụ thể
  - Cần có sự tương tác với người dùng trong quá trình hoạt động

## I.4. Phần mềm / ứng dụng phân tán (distributed software / application)

Tập các thành phần phần mềm/ tiến trình trong 1 mạng máy tính, cùng hoạt động nhằm phục vụ mục đích nhất định

Ví dụ về phần mềm / ứng dụng phân tán ?

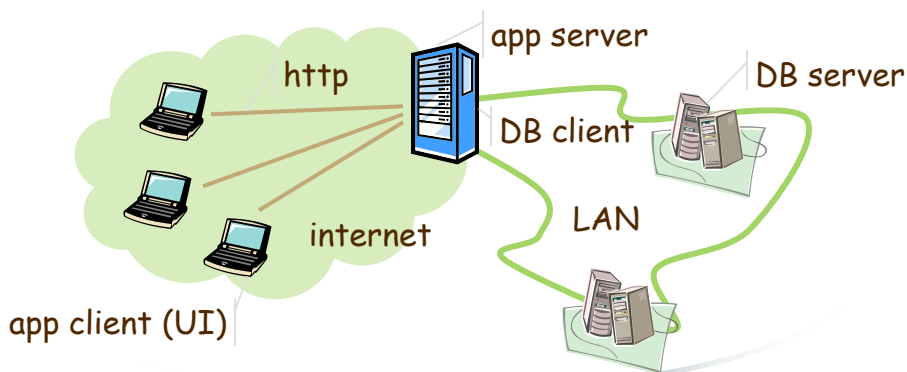
Phân tán trong phạm vi nào ?

Các đặc tính khác

- Khả năng mở rộng phạm vi (scalability) ?
- Tính trong suốt (transparency) ?
- Tương tác với người dùng?

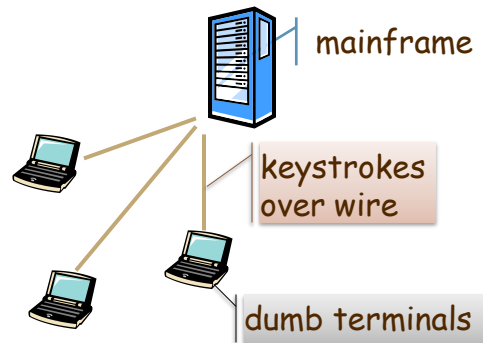
## Ví dụ: thương mại điện tử (eCommerce)

- Có phải là HPT hay không? Tại sao?
  - Các thành phần phân tán?
  - Kiểu tương tác giữa các thành phần này?



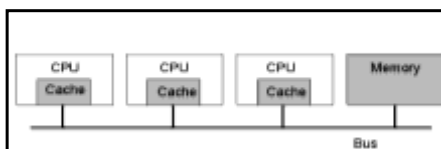
## Ví dụ: timesharing

- Có phải là HPT hay không? Tại sao?
  - Các thành phần phân tán?
  - Kiểu tương tác?

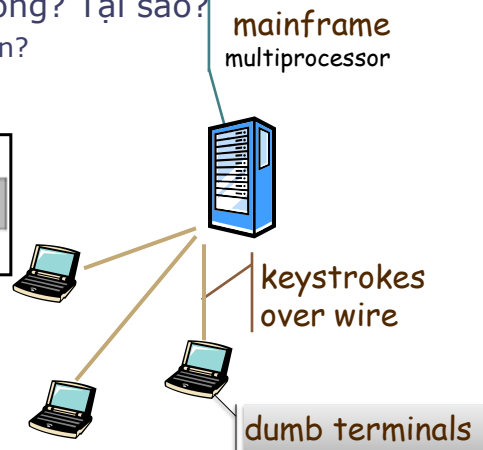


## Ví dụ: parallel computing

- Có phải là HPT hay không? Tại sao?
  - Các thành phần phân tán?
  - Kiểu tương tác?



Một số ứng dụng nổi bật ?



## II. PHÁT TRIỂN PHẦN MỀM PHÂN TÁN

Ví dụ: ứng dụng nhà sách trực tuyến  
Quy trình nghiệp vụ

- ✚ Online bookstore (e.g. in the World Wide Web)
  - ✚ Customers can connect their computer to your computer (web server):
    - Browse your inventory
    - Place orders
    - ...

## Ví dụ: ứng dụng nhà sách trực tuyến Thách thức

### ⊕ What if

- ⊕ Your customer uses a completely different hardware? (PC, MAC,...)
- ⊕ ... a different operating system? (Windows, Unix,...)
- ⊕ ... a different way of representing data? (ASCII, EBCDIC,...)

### ⊕ Or

- ⊕ You want to move your business and computers to the Caribbean (because of the weather)?
- ⊕ Your client moves to the Caribbean (more likely)?

## Ví dụ: ứng dụng nhà sách trực tuyến Thách thức

### ⊕ What if

- ⊕ Someone tries to break into your system to steal data?
- ⊕ ... sniffs for information?
- ⊕ ... your customer orders something and doesn't accept the delivery saying he didn't?

### ⊕ Or

- ⊕ You are so successful that millions of people are visiting your online store at the same time?

## Ví dụ: ứng dụng nhà sách trực tuyến Thách thức

- ✚ When building the system...
  - ✚ Do you want to write the whole software on your own (network, database,...)?
  - ✚ What about updates, new technologies?

## Ví dụ: ứng dụng nhà sách trực tuyến Thách thức

- ✚ What if
  - ✚ Two customers want to order the same item at the same time?
- ✚ Or
  - ✚ The database with your inventory information crashes?
  - ✚ Your customers computer crashes in the middle of an order?



## II.1. Thách thức

- ✚ Heterogeneity
  - ✚ Các thành phần không thuần nhất phải có khả năng tương tác với nhau
- ✚ Distribution transparency
  - ✚ Giấu người dùng sự phân tán nhiều nhất có thể
- ✚ Fault tolerance
  - ✚ Việc một thành phần trong hệ thống bị hỏng không làm hỏng cả hệ thống
- ✚ Scalability
  - ✚ Hệ thống có khả năng hoạt động hiệu quả khi số lượng người dùng tăng lên
  - ✚ Hiệu năng của hệ thống tăng lên khi bổ sung thêm tài nguyên
- ✚ Concurrency
  - ✚ Có khả năng cùng truy cập vào tài nguyên và chia sẻ tài nguyên
- ✚ Openness
  - ✚ Giao diện công khai, sẵn dùng và dễ dàng thêm các thành phần mới vào hệ thống
- ✚ Security
  - ✚ Hệ thống chỉ có thể được sử dụng theo cách định sẵn

## II.2. Mục đích

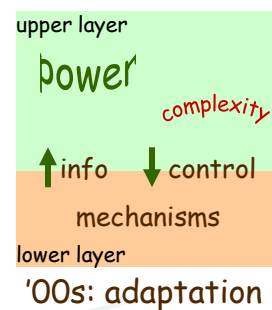
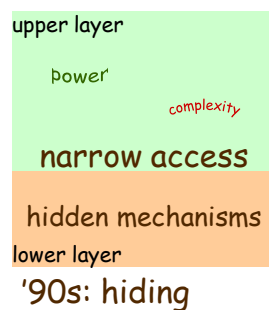
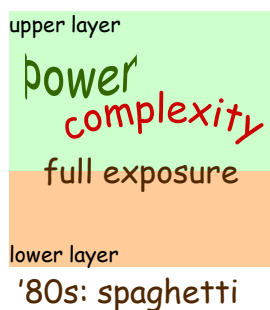
- Kết nối người dùng và các tài nguyên trong hệ thống
  - Kinh tế: mua 1 máy in
  - Cộng tác: chia sẻ tài nguyên
  - Cần đảm bảo an toàn (security)
- Che giấu sự phân tán (Transparency)
  - Các khía cạnh cần quan tâm?
- Tính mở (Openness)
  - Khả năng tương tác (Interoperability)
  - Khả chuyển (Portability)
  - Khả năng co dãn (Extensibility)
  - Phân biệt giữa chính sách và kỹ thuật
- Khả năng thay đổi phạm vi (Scalability)
- Nâng cao tính sẵn dùng (Availability)

## II.3. Cách tiếp cận

- Tận dụng thế mạnh sẵn có của hệ phân tán
- Độ trễ trong quá trình truyền thông
- Tích phối (coordination)
  - Chia sẻ tài nguyên và loại trừ lẫn nhau
  - Thứ tự thực hiện, deadlock, live-lock
  - Thời gian thực hiện
- Thích nghi với các thay đổi
  - Hỏng (failures), lỗi phần mềm (soft faults), tối ưu
  - Tìm kiếm và cấu hình các dịch vụ
  - Sự không thuần nhất và các phần mềm của bên thứ ba
  - Mở rộng phạm vi và tiến hóa
- Tính an toàn an ninh và tính riêng tư
  - Tin tưởng vào thiết bị, phần mềm, truyền thông và người dùng khác trong hệ thống
- Khả năng xử lý
- Dung lỗi, mở rộng phạm vi, tiến hóa
- Kiểm soát công khai, quyền ưu tiên

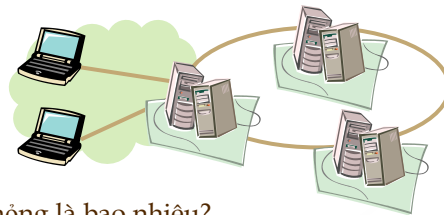
## II.3. Cách tiếp cận

- Phân tách chính sách / kỹ thuật
  - Giảm độ phức tạp trong quá trình phát triển phần mềm phân tán
  - Tăng khả năng thích nghi của PMPT



## Ví dụ: kỹ thuật tạo bản sao (replication) để dung lỗi

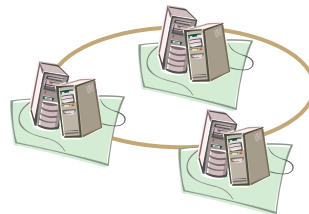
$F$  = probability server fails



- Xác suất để tất cả các server đều hỏng là bao nhiêu?
- Cần bao nhiêu bản sao để đảm bảo là tỉ lệ lỗi server luôn nhỏ hơn  $G$ ?

## Ví dụ: hạn chế của việc áp dụng kỹ thuật tạo bản sao

- Mua nhiều phần cứng hơn
- Chi phí quản lý
- Nâng cấp phần mềm
- Cân bằng tải
  - Quá tải
  - Phần mềm phức tạp hơn



## Ví dụ: kỹ thuật hiding nhằm mục tiêu giảm bớt sự phức tạp

[Tanenbaum] "transparency" goals:

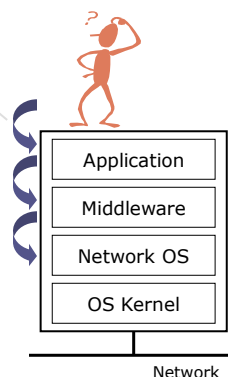
Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide the fact that several copies of a resource exist
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

## Ví dụ: kỹ thuật hiding nhằm mục tiêu giảm bớt sự phức tạp

hiding consists of:

(1) push handling complexity to a lower level

(2) implementing a one-size-fits-all solution



## Ví dụ: hạn chế của kỹ thuật hiding

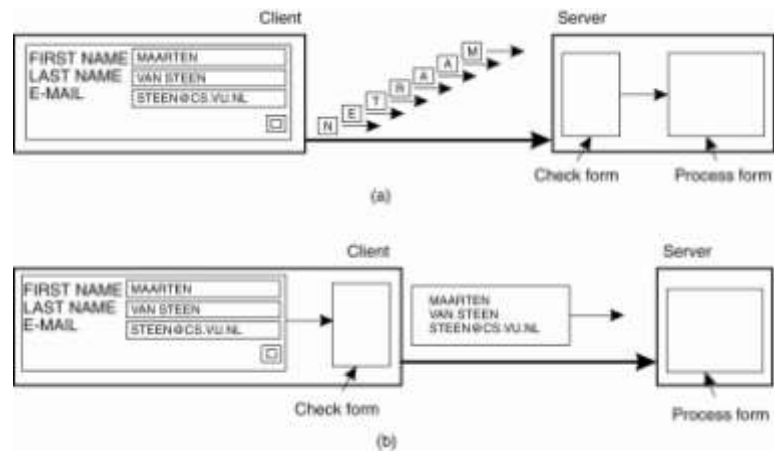
Transparency	Description
Access	<ul style="list-style-type: none"> <li>▪ conversion of complex formats, e.g. media</li> <li>▪ latency vs. fidelity of access, e.g. DBs, Web...</li> </ul>
Location	<ul style="list-style-type: none"> <li>▪ trusted hosts (security and privacy)</li> </ul>
Migration	<ul style="list-style-type: none"> <li>▪ different performance</li> </ul>
Relocation	<ul style="list-style-type: none"> <li>▪ different capabilities</li> <li>▪ different network access (bandwidth &amp; latency)</li> </ul>
Replication	<ul style="list-style-type: none"> <li>▪ select server based on QoS (e.g., mirrors)</li> </ul>
Concurrency	<ul style="list-style-type: none"> <li>▪ cannot hide sharing of resources: resources are consumed, data is modified by others</li> </ul>
Failure	<ul style="list-style-type: none"> <li>▪ unexplained behavior</li> </ul>
Persistence	<ul style="list-style-type: none"> <li>▪ someone needs to decide whether an object is persistent, and someone needs to commit it to disk</li> </ul>

## Ví dụ: Bài toán scaling

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

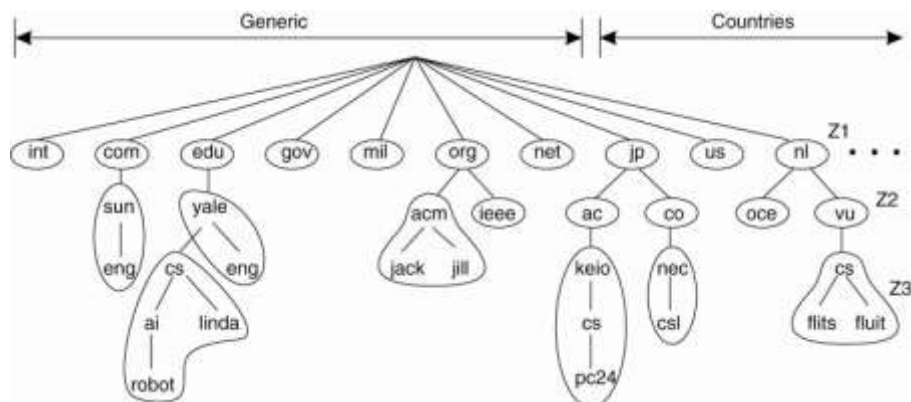
Examples of scalability limitations.

## Ví dụ: kỹ thuật scaling



The difference between letting (a) a server or (b) a client check forms as they are being filled.

## Ví dụ: kỹ thuật scaling



An example of dividing the DNS name space into zones.



## Ví dụ: kỹ thuật scaling

- Thúc đẩy các giải thuật phi tập trung (decentralized algorithms):
  - Các máy không cần cung cấp thông tin về trạng thái của hệ thống
  - Các máy tự đưa ra quyết định dựa trên những thông tin riêng
  - Nếu một máy bị lỗi, lỗi này không làm ảnh hưởng đến giải thuật
  - Không cần giả thiết là các máy trong hệ thống phải thực hiện đồng bộ

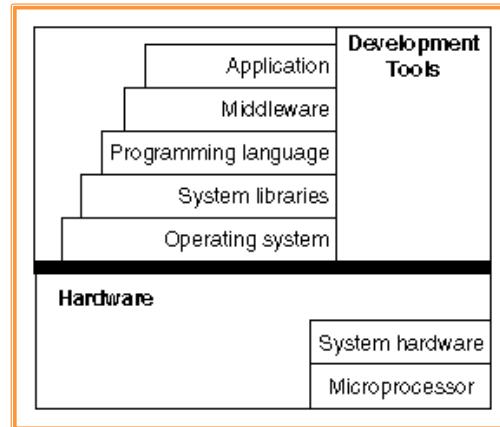


## III. CÁC NỀN TẢNG HỖ TRỢ QUÁ TRÌNH PHÁT TRIỂN PHẦN MỀM PHÂN TÁN



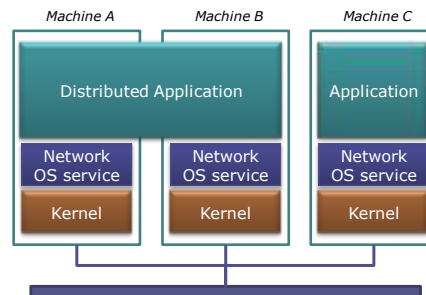
## Mở đầu

- Phát triển các phần mềm tốt là công việc khó khăn
- Phát triển các phần mềm phân tán tốt còn khó khăn hơn nữa: khó xây dựng PMPT trực tiếp trên các thiết bị phần cứng
- Cần có các software platforms trợ giúp việc phát triển PMPT
  - Hệ điều hành mạng
  - Hệ điều hành phân tán
  - Phần dẻo



## III.1. Hệ điều hành mạng (Network OS)

- Không che giấu sự phân tán
- Cung cấp các dịch vụ hỗ trợ phân tán:
  - remote login (ssh),
  - remote copying (scp)
  - distributed file systems (samba, nfs)
- Thích nghi tốt với các HPT khác nhau
  - mount a samba share (MS technology) on a Mac (Apple)
  - use ssh to connect from a PC to a Unix server

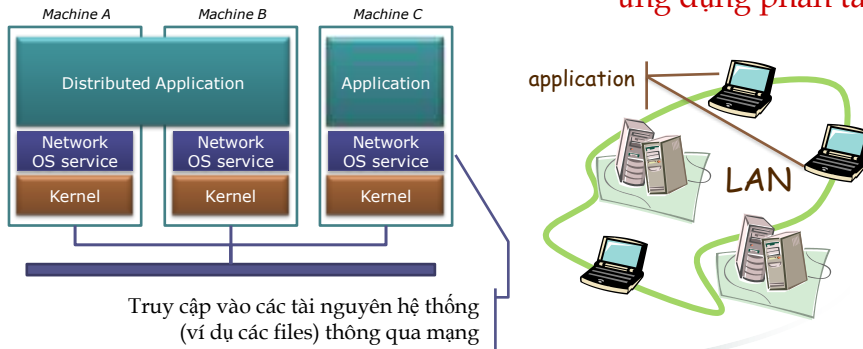




## Ví dụ: local networks

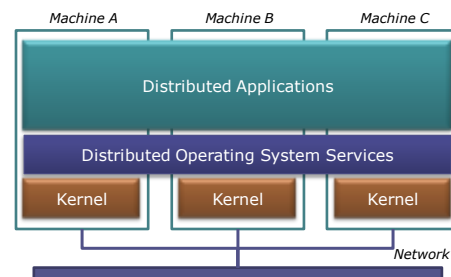
- Có phải là HPT hay không? Tại sao?
  - Các thành phần phân tán?
  - Kiểu tương tác?

Khó phát triển các ứng dụng phân tán !!!



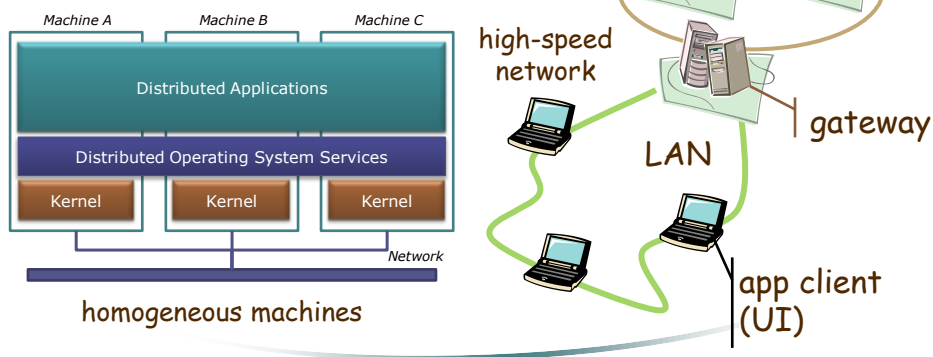
## III.2. Hệ điều hành phân tán (distributed OS)

- Che giấu hoàn toàn sự phân tán: hình ảnh của một hệ thống duy nhất
- Người dùng có cảm giác như đang sử dụng một máy tính có nhiều bộ xử lý (symmetric multi-processing, SMP)
- Có nhiều điểm tương đồng với HĐH hỗ trợ nhiều bộ xử lý



## Ví dụ: clusters

- Có phải là HPT hay không? Tại sao?
  - Các thành phần phân tán?
  - Kiểu tương tác?

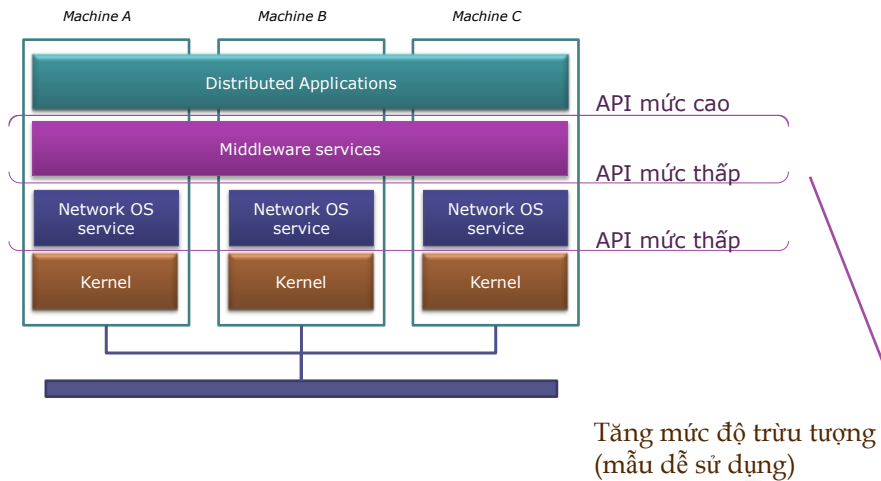


## III.3. Phần dẻo (middleware)

- Trong 1 HPT, các ứng dụng khó sử dụng trực tiếp các chức năng được cung cấp thông qua giao diện của hệ điều hành hay hệ truyền thông:
  - Tính không đồng nhất (heterogenous)
  - Sự phức tạp của các kỹ thuật bên dưới
- DOS thân thiện với người dùng nhưng khó mở rộng.
- NOS dễ mở rộng nhưng ít thân thiện với người dùng
- Cần quản lý các khả năng tương tác khác nhau và che giấu (dù chỉ phần nào) các điểm khác biệt này
- Giải pháp:
  - Thêm vào một tầng trung gian (phân tán) giữa các tầng ở mức thấp (hệ thống và truyền thông) và tầng ứng dụng → phần dẻo
  - Phần dẻo đóng vai trò tương tự như một siêu hệ điều hành trong 1 hệ phân tán

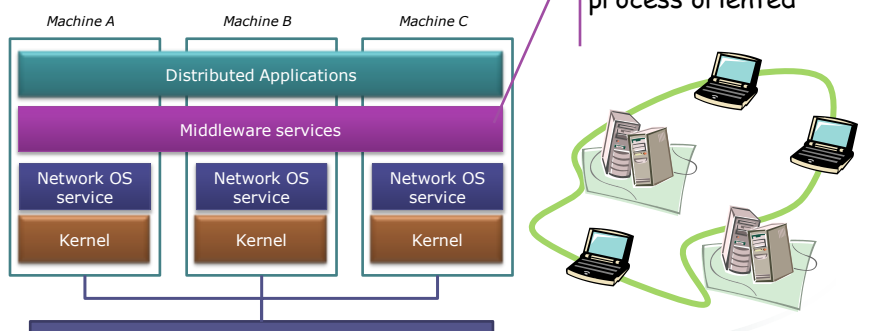


### III.3. Phần dero



### Ví dụ: middleware

- Có phải là HPT hay không? Tại sao?
  - Các thành phần phân tán?
  - Kiểu tương tác?



## Ví dụ: ubiquitous computing

- Ứng dụng phân tán ?



## Phân loại phần mềm do các software platforms cung cấp theo mục đích sử dụng

Hệ thống	Miêu tả	Mục đích
DOS	Hệ điều hành dành riêng cho các máy tính đa xử lý và các máy tính thuần nhất	Che giấu và quản lý tài nguyên phần cứng
NOS	Hệ điều hành cho các máy tính không thuần nhất, kết nối qua mạng LAN, WAN	Cung cấp dịch vụ tại chỗ cho các khách hàng ở xa
Middleware	Tầng xây dựng trên NOS, cài đặt các dịch vụ với mục tiêu chung	Che giấu sự phân tán



## So sánh các software platforms

Item	Distributed OS		Network OS	Middleware-based OS
	Multiproc.	Multicomp.		
Degree of transparency	Very High	High	Low	High
Same OS on all nodes	Yes	Yes	No	No
Number of copies of OS	1	N	N	N
Basis for communication	Shared memory	Messages	Files	Model specific
Resource management	Global, central	Global, distributed	Per node	Per node
Scalability	No	Moderately	Yes	Varies
Openness	Closed	Closed	Open	Open



## Kết luận

- Tổng quan về hệ phân tán, ứng dụng phân tán và phần mềm phân tán
  - Tại sao phải phân tán ?
  - Ví dụ thực tế của các hệ phân tán ?
  - Khả năng mở rộng/tiến hóa của các hệ thống này?
- Phát triển phần mềm phân tán : mục đích, cách tiếp cận và các vấn đề cần giải quyết
  - Tại sao việc phát triển phần mềm phân tán lại khó khăn hơn phát triển phần mềm thông thường ?
- Vai trò của phần dẻo, truyền thông và hệ điều hành cũng như các dịch vụ hỗ trợ khác trong quá trình phát triển phần mềm phân tán, ví dụ minh họa



## Tại sao phải phát triển các phần mềm phân tán

- Nhu cầu ứng dụng
  - Tích hợp các ứng dụng riêng rẽ đang có
  - Tích hợp các nguồn tài nguyên đang có
    - Lưới tính toán
    - Quản lý dữ liệu
  - Đưa tin học vào các lĩnh vực ứng dụng mới
    - Tích hợp các đối tượng trong cuộc sống thực
- Khả năng kỹ thuật
  - Chi phí và hiệu năng của máy tính và truyền thông