

## Bài tập tuần 1

Câu 1:

**a. Hai đặc điểm quan trọng nhất của hệ thống phân tán :**

- Là hệ thống các phần mềm máy tính độc lập
  - Các thành phần trong hệ thống phân tán được cài đặt trên các **máy tính khác nhau**, mỗi máy **có thể hoạt động độc lập**.
  - Mỗi máy có bộ xử lý, bộ nhớ, đồng hồ riêng, không chia sẻ tài nguyên vật lý như trong hệ thống tập trung.
- Các máy tính này phối hợp hoạt động với nhau như một thể thống nhất.
  - Dù độc lập, các phần tử **phối hợp với nhau qua việc trao đổi thông điệp** để xử lý bài toán như một hệ thống thống nhất.
  - Người dùng **có cảm giác như đang tương tác với một hệ thống đơn lẻ**, dù thực chất nhiều máy cùng tham gia xử lý.

**b. Ba lý do cơ bản khiến các ứng dụng phân tán phức tạp hơn so với các ứng dụng đơn lẻ :**

**1. Chia sẻ tài nguyên và dữ liệu phân tán**

- Dữ liệu không tập trung tại một nơi mà được **phân tán tại nhiều máy khác nhau** → phát sinh các vấn đề như:
  - Phân quyền truy cập
  - Đồng bộ dữ liệu
  - Bảo vệ chống sao chép trái phép

**2. Tối ưu hiệu năng và xử lý song song**

- Hệ thống phân tán tận dụng xử lý song song hoặc khả năng của các máy chuyên dụng → lập trình viên **phải thiết kế ứng dụng để chia nhỏ bài toán**, đảm bảo **cân bằng tải, đồng bộ dữ liệu và kết quả**.

**3. Yêu cầu về khả năng chịu lỗi và độ tin cậy**

- Hệ thống phải tiếp tục hoạt động ngay cả khi một hay nhiều máy con gặp sự cố → cần:
  - **Phát hiện lỗi**
  - **Chuyển hướng xử lý sang máy khác**
  - **Đảm bảo không gián đoạn dịch vụ**

Câu 2 :

**a. Phân tích các yếu tố phân cứng và mạng ảnh hưởng đến hiệu năng hệ thống phân tán**

♦ **Phân cứng**

- **CPU (bộ xử lý):**

- Nhiều bộ vi xử lý giúp tăng tốc độ xử lý song song.
- Trong hệ thống nhiều CPU chia sẻ bộ nhớ chung, nếu không có bộ nhớ đệm sẽ gây **lãng phí thời gian chờ**.
- Sử dụng **bộ nhớ đệm và bộ nhớ riêng** giúp tối ưu hóa truy cập và giảm xung đột kênh truyền.
- **Bộ nhớ (RAM):**
  - Truy cập bộ nhớ nội bộ trên một máy tính rất nhanh ( $\approx$  nanosecond).
  - Tuy nhiên, nếu cần truy cập bộ nhớ của máy tính khác qua mạng, **độ trễ tăng lên hàng triệu lần** ( $\approx$  vài chục ms).
- **Kênh truyền (bus nội bộ hoặc mạng):**
  - Là yếu tố giới hạn hiệu năng khi nhiều CPU chia sẻ cùng một bus.
  - **Kênh riêng hoặc bộ chuyển mạch** giúp giảm xung đột và tối ưu truyền dữ liệu.

#### ♦ Yếu tố mạng

- **Băng thông:**
  - Băng thông cao giúp trao đổi dữ liệu nhanh giữa các nút hệ thống.
  - Mạng LAN thường có băng thông lớn hơn WAN, ảnh hưởng lớn đến tốc độ trao đổi thông điệp.
- **Topology (cấu trúc mạng):**
  - Mạng **hình sao với switch**: tối ưu hiệu năng, mỗi máy một kênh riêng.
  - Mạng **bus hoặc dùng hub**: dễ xảy ra xung đột, giảm hiệu năng.

**b. Hệ điều hành phân tán (distributed OS) và hệ điều hành mạng (network OS) lại có yêu cầu khác nhau về quản lý tài nguyên vì :**

Tiêu chí	Distributed OS	Network OS
Mục tiêu chính	Làm cho nhiều máy hoạt động <b>như một hệ thống thống nhất</b>	Quản lý <b>các máy riêng biệt</b> trên cùng mạng
Quản lý tài nguyên	Phải <b>phân phối và đồng bộ tài nguyên</b> trên nhiều máy (CPU, bộ nhớ, thiết bị...)	Quản lý <b>tài nguyên cục bộ</b> trên từng máy riêng biệt
Giao tiếp	Ảnh chi tiết phân tán, <b>ứng dụng thấy như dùng 1 hệ thống</b>	Ứng dụng thường phải <b>biết rõ máy nào có tài nguyên gì</b>
Yêu cầu phức tạp hơn	Cao hơn nhiều do cần <b>đồng bộ thời gian, bảo mật, truyền thông xuyên máy</b>	Đơn giản hơn vì mỗi hệ điều hành <b>chỉ lo máy của mình</b>

Câu 3

**a. Nêu và so sánh ba loại hệ thống phân tán**

Tiêu chí	Điện toán phân tán	Thông tin phân tán	Lan tỏa phân tán
<b>Mục tiêu chính</b>	Tăng hiệu năng xử lý bằng cách phân chia tác vụ tính toán	Lưu trữ và truy vấn dữ liệu được phân tán trên nhiều máy	Tương tác linh hoạt, tự động giữa các thiết bị không đồng nhất, thường là thiết bị di động hoặc IoT
<b>Đặc điểm nổi bật</b>	- Chia nhỏ tác vụ để xử lý song song trên nhiều máy- Tập trung vào năng lực tính toán- Có thể là cụm hoặc lưới	- Dữ liệu lưu rải rác, cần đồng bộ và duy trì nhất quán khi ghi- Dùng phần mềm trung gian để truy vấn và cập nhật dữ liệu phân tán	- Thiết bị tự động cấu hình- Bao quát ngữ cảnh- Giao tiếp không dây, tính linh hoạt cao- Dữ liệu thường không lưu trữ tập trung
<b>Tính đồng nhất</b>	Có thể yêu cầu đồng nhất cao (cụm) hoặc chấp nhận dị biệt (lưới)	Không cần đồng nhất, chỉ cần	Dị biệt cao, đa dạng về thiết bị, OS, giao thức

		tương thích về truy vấn	
<b>Ví dụ</b>	Hệ thống Beowulf- MOSIX- Siêu máy tính ảo	Hệ thống CSDL phân tán- Web service kết nối nhiều nguồn dữ liệu	IoT trong nhà thông minh- Hệ thống chăm sóc sức khỏe từ xa- Mạng cảm biến môi trường

## b. Phân tích các lớp chính trong kiến trúc điện toán lưới

### ♦ Lớp Ứng dụng (Application Layer)

- Vai trò:
  - Cung cấp giao diện và chức năng cho người dùng cuối hoặc phần mềm sử dụng tài nguyên lưới.
  - Là nơi người dùng đưa ra yêu cầu xử lý bài toán, ví dụ: mô phỏng vật lý, xử lý hình ảnh lớn, AI phân tán...
- Đặc điểm:
  - Sử dụng dịch vụ lưới một cách trừu tượng, không cần biết chi tiết tầng dưới.

### ♦ Lớp Tiếp nhận (Reception Layer)

- Vai trò:
  - Xử lý yêu cầu từ tầng ứng dụng và phân tích nhu cầu truy cập tài nguyên.
  - Cung cấp các chức năng như:
    - Thăm dò tài nguyên
    - Lập lịch truy cập
    - Nhân bản dữ liệu
    - Định vị tài nguyên phù hợp
- Đặc điểm:
  - Không chuẩn hóa cố định, thường linh hoạt theo từng nền tảng lưới.

### ♦ Lớp Kết nối (Connectivity Layer)

- Vai trò:

- Cung cấp giao thức truyền thông bảo mật giữa các tài nguyên trong lưới.
- Xử lý các chức năng truyền, di chuyển hoặc truy cập tài nguyên.
- Thành phần:
  - Giao thức mạng
  - Cơ chế bảo mật: xác thực, phân quyền truy cập
- ◆ Lớp Tài nguyên (Resource Layer)
  - Vai trò:
    - Quản lý tài nguyên đơn lẻ như CPU, bộ nhớ, file, phần mềm...
    - Xác định trạng thái tài nguyên, khởi tạo và xử lý yêu cầu truy cập tài nguyên.
- ◆ Lớp Thực hiện (Execution Layer)
  - Vai trò:
    - Là tầng thấp nhất, thực thi yêu cầu thực tế trên tài nguyên cục bộ.
    - Bao gồm cơ chế giám sát, cập nhật trạng thái, thực hiện truy vấn, điều khiển truy cập tài nguyên.

#### Câu 4 :

##### a. “tính sẵn sàng” (availability) là mục tiêu quan trọng nhất của hệ thống phân tán vì :

Tính sẵn sàng là khả năng hệ thống đảm bảo người dùng có thể truy cập tài nguyên mọi lúc, mọi nơi, bất kể vị trí địa lý hay thời điểm. Đây là mục tiêu quan trọng nhất vì:

- Mục tiêu cốt lõi của hệ thống phân tán là kết nối người dùng với tài nguyên (máy in, dữ liệu, dịch vụ...) một cách liên tục và thuận tiện.
- Trong môi trường phân tán, tài nguyên nằm rải rác ở nhiều nơi, nên cần đảm bảo rằng người dùng không bị gián đoạn truy cập dù có lỗi phần cứng, mạng hay phần mềm.
- Tính sẵn sàng cao dẫn đến hiệu năng và trải nghiệm người dùng tốt hơn, đặc biệt với dịch vụ trực tuyến như thương mại điện tử, đào tạo, y tế...
- Bảo mật, lọc rác, riêng tư... đều phải được giải quyết mà không ảnh hưởng đến tính sẵn sàng, nghĩa là người dùng vẫn truy cập được tài nguyên mà không bị làm phiền hoặc chậm trễ.

Ví dụ: Trong một doanh nghiệp, nếu hệ thống máy in hoặc cơ sở dữ liệu bị gián đoạn, toàn bộ quy trình làm việc có thể ngừng lại — đây là thất bại nghiêm trọng của tính sẵn sàng.

#### **b. Ba hình thức "tính trong suốt" và ví dụ minh họa**

- Trong suốt về truy nhập (Access Transparency)
  - Ý nghĩa: Người dùng không cần biết cách thức truy cập tài nguyên, dù tài nguyên đó nằm ở máy nào, có định dạng gì, giao thức gì...
  - Mục tiêu: Che giấu sự khác biệt trong cách biểu diễn dữ liệu và giao thức truy cập.
  - Ví dụ: Một người dùng truy cập tập tin từ mạng như đang mở tập tin cục bộ trên máy mình, không cần quan tâm nó nằm trên hệ điều hành Linux hay Windows, hoặc đang dùng FTP hay HTTP.
- 
- Trong suốt về vị trí (Location Transparency)
  - Ý nghĩa: Người dùng không cần biết tài nguyên đặt ở đâu trong hệ thống.
  - Mục tiêu: Ẩn vị trí địa lý của tài nguyên để tạo cảm giác như toàn bộ hệ thống nằm trên một máy.
  - Ví dụ: Truy cập vào trang web [www.ptit.edu.vn](http://www.ptit.edu.vn) mà không cần biết máy chủ đang đặt ở Hà Nội, TP.HCM hay quốc gia nào khác.
- Trong suốt về lỗi (Failure Transparency)
  - Ý nghĩa: Người dùng không thấy hoặc bị ảnh hưởng bởi lỗi, như sự cố phần cứng, mất kết nối, hoặc lỗi phần mềm.
  - Mục tiêu: Hệ thống tự phục hồi hoặc che giấu lỗi để tiếp tục cung cấp dịch vụ.
  - Ví dụ: Một dịch vụ lưu trữ có nhiều bản sao dữ liệu, nếu một máy chủ bị hỏng, hệ thống tự động chuyển sang bản sao khác mà người dùng không hề nhận thấy có sự cố nào.

○

#### **c. Mối quan hệ giữa "tính mở" (openness) và "khả năng tương tác" (interoperability)**

Tính mở:

- Là khả năng hệ thống kết nối với các hệ thống khác, dễ dàng nâng cấp, mở rộng, tương thích với nhiều nền tảng phần cứng và phần mềm.
- Được đảm bảo nhờ việc tuân thủ các chuẩn giao tiếp công khai (protocols, APIs, định dạng dữ liệu...).

Khả năng tương tác:

- Là khả năng mà các thành phần đến từ nhà cung cấp khác nhau có thể làm việc cùng nhau.
- Được hiện thực thông qua việc sử dụng các giao diện mô tả rõ cú pháp và ngữ nghĩa (ví dụ: REST, SOAP, WSDL, IDL...).

Mối quan hệ:

- Tính mở là điều kiện tiên quyết để đạt được khả năng tương tác.
- Hệ thống mở cần phải:
  - Tuân thủ các tiêu chuẩn chung (ví dụ: TCP/IP, HTTP, SCORM...),
  - Cung cấp mô tả giao diện rõ ràng,
  - Hỗ trợ các cách mở rộng, thay thế thành phần mà không ảnh hưởng đến phần còn lại của hệ thống.

Ví dụ:

- Một hệ thống thương mại điện tử có tính mở nếu cho phép tích hợp với nhiều cổng thanh toán (PayPal, ZaloPay, VNPay...) thông qua các API chuẩn hóa.
- Nhờ đó, hệ thống này có khả năng tương tác cao, dễ dàng thay thế hoặc bổ sung cổng thanh toán mới mà không cần sửa lại toàn bộ hệ thống.

**Câu 5:**

**a. So sánh ưu – nhược điểm của kiến trúc phân cấp và kiến trúc ngang hàng trong hệ thống phân tán.**

Tiêu chí	Kiến trúc phân cấp (hierarchical)	Kiến trúc ngang hàng (peer-to-peer)
Cấu trúc tổ chức	Dạng cây, có nút cha – nút con, thể hiện quan hệ phụ thuộc	Không có phân cấp, các nút bình đẳng
Ưu điểm	- Dễ quản lý, giám sát và mở	- Tính linh hoạt cao, mỗi nút có

	rộng hệ thống. - Phù hợp với việc phân chia chức năng theo tầng. - Lan truyền thông tin hiệu quả theo cây.	thể vừa là khách vừa là chủ. - Không có điểm lỗi trung tâm. - Có thể hoạt động ngẫu nhiên và độc lập. - Phân phối tải đều hơn.
Nhược điểm	- Phụ thuộc mạnh vào các nút cha, nếu nút cha bị lỗi thì cả nhánh mất dịch vụ. - Có thể tạo nút cổ chai nếu mức truy cập quá tải.	- Khó quản lý và điều phối, đặc biệt khi có nhiều thành viên. - Truy vấn dữ liệu có thể mất thời gian (đặc biệt với mạng phi cấu trúc).
Tính chịu lỗi	Kém hơn do phụ thuộc vào nút cha	Cao hơn vì không có điểm lỗi trung tâm
Hiệu quả tìm kiếm	Tìm kiếm nhanh theo hướng cây (nếu tổ chức tốt)	- Có thể hiệu quả với mạng có cấu trúc (DHT) - Nhưng kém hiệu quả nếu mạng phi cấu trúc
Tính mở rộng	Tốt nhưng cần cẩn trọng khi mở rộng cây quá nhiều tầng	Rất cao vì có thể thêm hoặc gỡ thành viên động

**b. Trình bày bốn mô hình hệ thống phân tán (phân tầng, đối tượng phân tán, kênh sự kiện, dữ liệu tập trung) và cho ví dụ ứng dụng điển hình cho mỗi mô hình.**

**1. Mô hình phân tầng (Layered Model)**

Đặc điểm:

- Hệ thống được chia thành các tầng có chức năng riêng biệt.
- Giao tiếp chỉ diễn ra giữa các tầng liền kề.
- Dễ bảo trì, mở rộng, nâng cấp từng tầng mà không ảnh hưởng đến tầng khác.
- Tầng tính chuyên môn hóa và đảm bảo tính trong suốt.

Ví dụ điển hình:

- Ứng dụng web 3 tầng (Three-tier Web App):
  - Tầng 1: Giao diện người dùng (UI) – trình duyệt web.



- Tầng 2: Tầng xử lý nghiệp vụ – ứng dụng server (Java/Spring, Node.js, .NET...).
- Tầng 3: Tầng dữ liệu – cơ sở dữ liệu (MySQL, PostgreSQL...).

Hệ thống sử dụng: Trang web tìm kiếm như Google Search (giao diện nhập từ khóa → tầng xử lý truy vấn → tầng dữ liệu lưu chỉ mục trang web).

## 2. Mô hình đối tượng phân tán (Distributed Object Model)

Đặc điểm:

- Mỗi thành phần được biểu diễn dưới dạng đối tượng, trao đổi với nhau qua gọi thủ tục từ xa (RPC) hoặc gọi phương thức từ xa (RMI).
- Có thể triển khai mô hình client/server.
- Ràng buộc giữa các đối tượng chặt hơn, phù hợp cho xử lý thời gian thực.

Ví dụ điển hình:

- DCOM (Distributed COM) của Microsoft, CORBA, hoặc Java RMI.
- Hệ thống quản lý phân tán ngân hàng, nơi mỗi đối tượng là một tài khoản hoặc dịch vụ (chuyển tiền, kiểm tra số dư), hoạt động như một phần tử độc lập nhưng có thể gọi nhau.

## 3. Mô hình kênh sự kiện (Event Channel Model)

Đặc điểm:

- Các thành phần không tương tác trực tiếp mà giao tiếp qua một kênh trung gian sự kiện.
- Đảm bảo ràng buộc lỏng giữa các thành phần, giúp hệ thống linh hoạt, dễ mở rộng và hỗ trợ dự phòng nóng.
- Mỗi tiến trình chỉ cần đăng ký sự kiện và lắng nghe, không cần biết bên gửi/nhận cụ thể.

Ví dụ điển hình:

- Hệ thống publish-subscribe (Pub/Sub) như trong Apache Kafka hoặc RabbitMQ.

- Hệ thống giám sát mạng hoặc IoT: cảm biến gửi dữ liệu đến kênh sự kiện, các tiến trình xử lý chỉ nhận sự kiện mà mình đã đăng ký.

#### 4. Mô hình dữ liệu tập trung (Shared Data Model)

Đặc điểm:

- Các thành phần trong hệ thống giao tiếp thông qua một kho dữ liệu dùng chung.
- Không yêu cầu tiến trình cung cấp dịch vụ luôn hoạt động đồng thời.
- Giúp chia sẻ dữ liệu lớn, độc lập thành phần và thuận tiện xử lý không đồng bộ.

Ví dụ điển hình:

- Hệ thống chia sẻ tập tin như Dropbox, Google Drive.
- Không gian chia sẻ dữ liệu như hệ thống Blackboard trong AI hoặc Tuple Space (Linda): tiến trình ghi dữ liệu vào kho, tiến trình khác đọc khi cần.

#### c. Vai trò của phần mềm trung gian (middleware) trong kiến trúc khách-chủ phân tán, và liệt kê ba tính năng chính mà nó cung cấp.

Phần mềm trung gian (middleware) đóng vai trò là **tầng trung gian** giữa tiến trình máy khách và máy chủ trong hệ thống phân tán. Nó giúp **đơn giản hóa việc thiết kế hệ thống**, đảm bảo **tính trong suốt trong phân phối dữ liệu, xử lý và điều khiển**, đồng thời hỗ trợ thiết lập, giám sát, bảo mật và đồng bộ các tương tác giữa các thành phần phân tán.

**Ba tính năng chính mà phần mềm trung gian cung cấp:**

1. **Thiết lập và quản lý phiên làm việc giữa các tiến trình** (session management), đảm bảo kết nối ổn định giữa máy khách và máy chủ.
2. **Bảo mật và xử lý lỗi**, bao gồm kiểm tra quyền truy cập, mã hóa dữ liệu, xử lý lỗi và đảm bảo an toàn thông tin.
3. **Chuyển tiếp và điều phối yêu cầu**, tức là nhận yêu cầu từ máy khách, gọi tiến trình phù hợp trên máy chủ và trả kết quả về máy khách một cách đồng bộ hoặc không đồng bộ.

### Câu 6 :

#### a. Phân loại ba loại dịch vụ trong SOA (cơ bản, tích hợp, quy trình) kèm ví dụ điển hình cho mỗi loại.

##### 1. Dịch vụ cơ bản (Basic Services)

Định nghĩa:

Là những dịch vụ kỹ thuật tái sử dụng, phục vụ cho các chức năng cơ bản của hệ thống, không gắn trực tiếp với nghiệp vụ mà hỗ trợ cho các dịch vụ nghiệp vụ khác.

Ví dụ điển hình:

- Dịch vụ đăng nhập, xác thực người dùng,
- Dịch vụ truy xuất cơ sở dữ liệu,
- Dịch vụ mã hóa dữ liệu.

##### 2. Dịch vụ tích hợp (Integration Services)

Định nghĩa:

Là các dịch vụ đóng vai trò kết nối và điều phối giữa nhiều dịch vụ cơ bản hoặc nghiệp vụ khác nhau. Chúng thực hiện truyền dữ liệu, chuyển đổi định dạng, hoặc đồng bộ hóa giữa các hệ thống khác nhau.

Ví dụ điển hình:

- Dịch vụ đồng bộ dữ liệu khách hàng giữa hệ thống bán hàng và hệ thống chăm sóc khách hàng,
- Dịch vụ chuyển đổi dữ liệu từ XML sang JSON giữa các dịch vụ không đồng nhất,
- Dịch vụ thông dịch giữa các miền dữ liệu khác nhau trong một doanh nghiệp lớn.

##### 3. Dịch vụ quy trình (Process Services)

Định nghĩa:

Là các dịch vụ phối hợp nhiều dịch vụ khác nhau (cơ bản và tích hợp) để thực hiện một quy trình nghiệp vụ hoàn chỉnh. Chúng thường thể hiện luồng công việc (workflow) cụ thể.

Ví dụ điển hình:

- Dịch vụ xử lý đơn hàng: kết hợp dịch vụ giỏ hàng, kiểm tra tồn kho, thanh toán, gửi thông báo và cập nhật đơn hàng,
- Dịch vụ duyệt nghỉ phép: phối hợp giữa nộp đơn, kiểm duyệt, phản hồi và cập nhật trạng thái đơn.

**b. Trình bày vòng đời của một dịch vụ SOA, từ giai đoạn phát triển đến vận hành sản xuất, và những thách thức chính ở mỗi giai đoạn.**

## **1. Giai đoạn phát triển (Development Phase)**

### **Mục tiêu chính:**

- Xác định các **quy trình nghiệp vụ** cần số hóa.
- **Thiết kế, mô hình hóa và xây dựng** các dịch vụ phù hợp với quy trình.
- Xác định **giao diện dịch vụ** và chuẩn hóa việc giao tiếp (SOAP, WSDL, HTTP...).

### **Các hoạt động chính:**

- Phân tích quy trình nghiệp vụ → xác định dịch vụ cần thiết.
- Thiết kế hợp đồng dịch vụ (interface, policy, metadata).
- Đóng gói chức năng theo mô hình module & tách biệt giao diện/triển khai.
- Kiểm thử chức năng & tích hợp.

### **Thách thức chính:**

- **Xác định đúng dịch vụ** để tách riêng và tái sử dụng.
- **Giao tiếp không đồng nhất** giữa các hệ thống khác nền tảng.
- **Thiết kế hợp đồng dịch vụ** đảm bảo lỏng lẻo nhưng rõ ràng.
- Đồng bộ dữ liệu và định dạng giữa các miền dịch vụ.

## **2. Giai đoạn vận hành sản xuất (Production Phase)**

### **Mục tiêu chính:**

- Đưa dịch vụ vào **hoạt động thực tế** trong môi trường doanh nghiệp.
- Đảm bảo **ổn định, hiệu năng, bảo mật và khả năng mở rộng**.

#### **Các hoạt động chính:**

- Triển khai dịch vụ trên trục dịch vụ doanh nghiệp.
- Giám sát & quản lý trạng thái dịch vụ.
- Cập nhật, nâng cấp hoặc sửa đổi khi có yêu cầu mới.
- Kiểm thử hiệu suất, bảo mật và khả năng tương tác.

#### **Thách thức chính:**

- **Thay đổi dịch vụ khi hệ thống đang vận hành** mà không ảnh hưởng các ứng dụng đang dùng dịch vụ đó.
- **Quản lý lỗi, bảo trì và tương thích ngược** giữa các phiên bản dịch vụ.
- **Đảm bảo bảo mật** cho các giao tiếp giữa dịch vụ.
- Khó khăn khi phải **tạm ngừng dịch vụ để cập nhật** trong trường hợp nghiêm trọng.