

Câu 1

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

a. Tính minh bạch (Transparency):

Là khả năng che giấu sự phức tạp của hệ thống phân tán với người dùng.

Ví dụ: người dùng không cần biết dữ liệu nằm trên server nào.

b. Tính mở (Openness):

Hệ thống hỗ trợ khả năng mở rộng và tương tác với các hệ thống khác thông qua các giao diện tiêu chuẩn (API, giao thức).

2. Ba lý do khiến ứng dụng phân tán phức tạp hơn:

a. Giao tiếp mạng không đáng tin cậy:

Không giống trong hệ thống đơn, việc truyền dữ liệu giữa các node có thể mất mát, trễ hoặc bị lỗi

b. Đồng bộ hóa và nhất quán dữ liệu khó hơn:

Dữ liệu được chia sẻ giữa nhiều nút nên cần xử lý các vấn đề về đồng bộ, mâu thuẫn dữ liệu.

c. Quản lý lỗi phức tạp:

Khó xác định lỗi là do phần mềm, phần cứng hay mạng.

Câu 2

1. Phân tích phần cứng và mạng ảnh hưởng hiệu năng:

Yếu tố	Ảnh hưởng
CPU	Tốc độ xử lý tác vụ tại mỗi node
Bộ nhớ	Khả năng lưu trữ cache tạm thời, ảnh hưởng đến tốc độ
Kênh truyền (Bus)	Ảnh hưởng tốc độ truy cập tài nguyên nội bộ
Băng thông	Giới hạn lượng dữ liệu có thể trao đổi giữa các node
Topology mạng	Ảnh hưởng độ trễ và mức chịu lỗi (vd: mesh nhanh hơn star)

2. Khác biệt giữa Distributed OS và Network OS:

Tiêu chí	Distributed OS	Network OS
Quản lý tài nguyên dùng điều khiển	Đồng bộ và tự động phân phối	Mỗi hệ thống tự quản, người

Mức độ tích hợp riêng biệt	Rất cao (như 1 hệ điều hành duy nhất)	Thấp, từng hệ điều hành
Tính minh bạch	Cao	Thấp

Câu 3

1. So sánh 3 loại hệ thống phân tán:

Loại	Đặc điểm	Ví dụ
Điện toán phân tán	Tập trung vào chia nhỏ tính toán	Hệ thống HPC
Thông tin phân tán	Dữ liệu phân tán trên nhiều node	Google Drive, CDN
Lan tỏa (Ubiquitous)	Thiết bị thông minh, cảm biến hoạt động liên tục	IoT, Smart Home

2. Các lớp chính trong kiến trúc điện toán lưới:

Resource layer: quản lý tài nguyên phần cứng (CPU, bộ nhớ,...)

Middleware layer: tạo môi trường thống nhất, điều phối, lịch biểu tài nguyên

Application layer: các ứng dụng cuối cùng sử dụng hệ thống lưới

Câu 4

1. Tính sẵn sàng (Availability) quan trọng vì:

Hệ thống phân tán phục vụ nhiều người, nếu 1 node hỏng mà hệ thống dừng, sẽ ảnh hưởng lớn → tính sẵn sàng đảm bảo liên tục.

2. Ba hình thức “tính trong suốt”:

Loại	Ý nghĩa
Trong suốt về truy cập	Người dùng không biết đang dùng tài nguyên cục bộ hay từ xa
Trong suốt về vị trí	Không cần biết file hay dịch vụ nằm ở đâu
Trong suốt về lỗi	Hệ thống tự động xử lý lỗi mà không ảnh hưởng đến người dùng

3. Môi quan hệ tính mở và khả năng tương tác:

Tính mở: hệ thống sử dụng giao thức, API tiêu chuẩn

Khả năng tương tác (interoperability): nhờ vào tính mở, các hệ thống khác nhau vẫn có thể hoạt động cùng nhau.

Câu 5

1. So sánh kiến trúc phân cấp và ngang hàng:

Tiêu chí	Phân cấp (Client-Server)	Ngang hàng (P2P)
Quản lý	Dễ, tập trung	Khó, phân tán
Hiệu suất	Có thể bị nghẽn	Phân tải tốt
Tin cậy	Server là điểm yếu	Khả năng chịu lỗi cao hơn

2. Bốn mô hình hệ thống phân tán:

Mô hình	Mô tả	Ví dụ
Phân tầng (Layered)	Tách thành các lớp (UI, Logic, DB)	Web 3-tier
Đối tượng phân tán	Gọi phương thức từ xa (RMI, CORBA)	Java RMI
Kênh sự kiện	Các thành phần phản ứng khi có sự kiện	Notification system
Dữ liệu tập trung	Dữ liệu được lưu ở nơi cố định	Dropbox, Google Drive

3. Vai trò của middleware và 3 tính năng chính:

Vai trò:

Kết nối client – server, ẩn sự phức tạp hệ thống

3 tính năng chính:

Giao tiếp giữa các thành phần

Quản lý tài nguyên & truy cập

Đảm bảo an toàn và bảo mật

Câu 6

1. Ba loại dịch vụ trong SOA:

Loại	Mô tả	Ví dụ
Dịch vụ cơ bản	Chức năng riêng lẻ	Kiểm tra tồn kho

Dịch vụ tích hợp Gộp nhiều dịch vụ cơ bản

Quản lý đơn hàng

Dịch vụ quy trình Điều phối workflow
hàng

Đặt hàng trực tuyến từ đặt → giao

2. Vòng đời dịch vụ SOA:

Phân tích & Thiết kế:

Xác định yêu cầu và mô hình hóa dịch vụ

Thách thức: Hiểu đúng nghiệp vụ

Phát triển & Triển khai:

Viết mã, triển khai lên server

Thách thức: Kiểm thử tích hợp phức tạp

Đăng ký & Công bố:

Đăng dịch vụ lên Service Registry

Thách thức: Quản lý phiên bản

Vận hành & Bảo trì:

Theo dõi, cập nhật, sửa lỗi dịch vụ

Thách thức: Bảo mật, hiệu suất