**BÀI TẬP TRÊN LỚP**

**MÔN HỌC: HỆ PHÂN TÁN**

**CHƯƠNG 1: TỔNG QUAN VÀ KIẾN TRÚC HỆ PHÂN TÁN**

HỌ TÊN SV: Nguyễn Đức Thắng

MSSV: 20225089

MÃ LỚP: 157542

MÃ HỌC PHẦN: IT4611

**Câu hỏi 1:** Em hãy nêu thêm 2 ví dụ về dịch vụ được coi là Hệ Phân Tán (ngoài 2 ví dụ WWW và Email đã trình bày trên lớp). Dựa vào định nghĩa, giải thích tại sao chúng được coi là Hệ Phân Tán.

Hệ thống cơ sở dữ liệu:

+ Gồm các máy tính có thể khác nhau về cấu hình và phần cứng phụ trách lưu trữ dữ liệu

+ Kết nối với nhau qua hệ thống mạng

+ Cung cấp dịch vụ một cách thống nhất: đảm bảo khi thực hiện truy xuất dữ liệu có thể truy xuất tại tất cả máy tính

Hệ thống huấn luyện các mô hình AI

+ Các thiết bị xử lý tính toán khác nhau nhưng cùng sử dụng để tính toán chô mô hình

+ Kết nối qua hệ thống dây hoặc mạng Internet

+ Các nhà nghiên cứu không cần quá quan tâm đến chi tiết khi xây dựng mô hình

**Câu hỏi 2:** Tại sao nói tính chia sẻ tài nguyên của Hệ Phân Tán có khả năng: Giảm chi phí, tăng tính sẵn sàng và hỗ trợ làm việc nhóm? Tuy nhiên lại tăng rủi ro về an toàn thông tin? Giải thích.

* Giảm chi phí: Các máy tính trong hệ phân tán có thể tận dụng tối đa tài nguyên của nhau, giúp giảm chi phí cho việc đầu tư tài nguyên.
* Tăng tính sẵn sàng: Người sử dụng có thể kết nối, truy cập các tài nguyên từ xa và từ các nút khác nhau làm giảm thời gian phản hồi và tăng tính sẵn sàng của hệ thống.
* Hỗ trợ làm việc nhóm: Những người sử dụng trong cùng nhóm dù cho có sử dụng các thiết bị có cấu hình phần cứng hay phần mềm khác nhau cũng có thể cùng thao tác với những tài nguyên được chia sẻ, thuận tiện hơn việc sao chép hay in ấn.
* Tăng rủi ro an toàn thông tin: Nguồn tài nguyên chia sẻ có thể chưa được xác thực (thông tin sai, mạo danh, …). Việc chia sẻ đồng nghĩa tăng số lượng kết nối mạng dẫn đến tăng lỗ hổng. Ngoài ra việc chia sẻ thông tin có thể dẫn đến việc vô tình để lộ những thông tin riêng tư.

**Câu hỏi 3:** Liên quan đến tính trong suốt, giải thích tại sao nhà quản trị hệ thống phải xem xét việc cân bằng giữa hiệu năng và độ trong suốt? Đưa ra ví dụ cụ thể để giải thích.

Khi hiệu năng càng cao thì đọ trong suốt cảng thấp vì vậy, mỗi hệ thống sẽ có tỉ lệ cân bằng phù hợp nên cần phân tích kỹ.

Ví dụ:

Khi đồng bộ dữ liệu giữa các máy chủ trong một hệ thống phân tán, khi nhiều máy chủ được đặt ở khắp nơi trên thế giới và cần được đồng bọ dữ liệu theo thời gian thực. Điều này sẽ gây tiêu tốn đáng kể tài nguyên hệ thống. Nếu điều này quá tốt thì hiệu năng hệ thống sẽ không được cao, còn nếu điều này quá tệ thì người dùng sex nhận ra tính không trong suốt của hệ thống.

**Câu hỏi 4: (không có)**

**Câu hỏi 5:** So sánh 2 kiểu HĐH DOS và NOS. Giải thích tại sao việc sử dụng Middleware là sự kết hợp ưu điểm của cả 2 mô hình trên.

So sánh 2 kiểu HĐH DOS và NOS

|  |  |  |  |
| --- | --- | --- | --- |
|  | DOS | | NOS |
| Multiprocessor | Multicomputer |
| Mức độ trong suốt | Rất cao | Cao | Thấp |
| Một HĐH trên các nút | Có | Có | Không |
| Số lượng bản sao của HĐH | Một | Nhiều | Nhiều |
| Trao đổi thông tin | Bộ nhớ chia sẻ | Chuyển thông báo | Tệp |
| Quản lý tài nguyên | Toàn cục tập trung | Toàn cục phân tán | Theo nút |
| Tính co giãn | Không | Có thể | Có |
| Tính mở | Đóng | Đóng | Mở |

Middleware kế thừa ưu điểm của cả DOS và NOS

**Tính trong suốt như DOS**

Middleware che giấu sự phức tạp của mạng và phân tán, cho phép ứng dụng truy cập tài nguyên như đang chạy trên một máy duy nhất.

**Linh hoạt và khả năng mở rộng như NOS**

Middleware không yêu cầu thay đổi hệ điều hành gốc, chạy trên các NOS sẵn có. Điều này giúp hệ thống dễ mở rộng và tích hợp các máy tính đa dạng.

**Câu hỏi 6:** Trong mô hình kiến trúc phân tầng OSI của Mạng máy tính, hãy trình bày tóm tắt chức năng của từng tầng. Lấy ví dụ cụ thể khi chúng ta thay đổi/cập nhật một tầng bất kỳ thì không ảnh hưởng đến hoạt động của các tầng khác.

Kiến trúc phân tâng OSI gồm 7 tầng

1. Tầng Vật lý (Physical Layer)

* Chức năng: Truyền tải dữ liệu thô qua phương tiện vật lý (cáp, sóng radio, ánh sáng).
* Ví dụ: Cáp Ethernet, cáp quang, tín hiệu Wi-Fi.

2. Tầng Liên kết dữ liệu (Data Link Layer)

* Chức năng: Đảm bảo truyền dữ liệu không lỗi giữa các thiết bị trên cùng một mạng (địa chỉ MAC, kiểm soát lỗi).
* Ví dụ: Giao thức Ethernet, Wi-Fi (802.11), PPP.

3. Tầng Mạng (Network Layer)

* Chức năng: Định tuyến dữ liệu giữa các mạng khác nhau (địa chỉ IP, định tuyến).
* Ví dụ: Giao thức IP, ICMP, ARP, Router.

4. Tầng Giao vận (Transport Layer)

* Chức năng: Đảm bảo truyền dữ liệu tin cậy giữa các ứng dụng (kiểm soát lỗi, điều khiển luồng).
* Ví dụ: TCP (đáng tin cậy), UDP (không đáng tin cậy).

5. Tầng Phiên (Session Layer)

* Chức năng: Quản lý phiên giao tiếp giữa các ứng dụng (thiết lập, duy trì, kết thúc phiên).
* Ví dụ: NetBIOS, RPC.

6. Tầng Trình diễn (Presentation Layer)

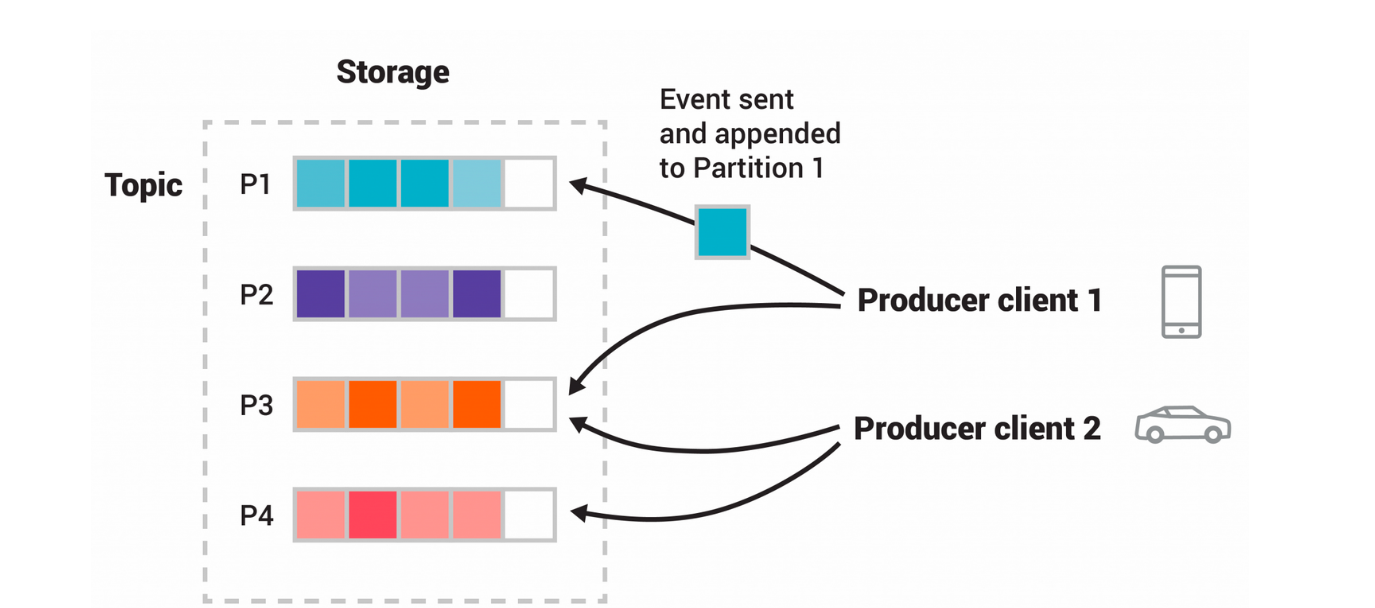
* Chức năng: Chuyển đổi dữ liệu sang định dạng phù hợp (mã hóa, nén, giải nén).
* Ví dụ: SSL/TLS (mã hóa), JPEG, MPEG.

7. Tầng Ứng dụng (Application Layer)

* Chức năng: Cung cấp giao diện cho người dùng và ứng dụng (giao thức ứng dụng).
* Ví dụ: HTTP, FTP, SMTP, DNS.

Câu hỏi 7: Cho ví dụ và phân tích một mô hình kiến trúc thuê bao/xuất bản (publish/subscribe).

Ví dụ về mô hình **Publish/Subscribe là Apache Kafka**

****

Apache Kafka sử dụng mô hình Pub/Sub để tách biệt các ứng dụng gửi tin nhắn (producers) và nhận tin nhắn (consumers), cho phép giao tiếp không đồng bộ. Điều này giúp giảm độ trễ và trở ngại so với giao tiếp đồng bộ trực tiếp giữa các ứng dụng.

Kiến trúc nhắn tin

* Tách biệt ứng dụng: Producers và consumers không phụ thuộc lẫn nhau, tập trung vào sự kiện (message) như một thực thể dựa trên thời gian thay vì vai trò cố định (như client-server).
* Publishing: Tin nhắn được "xuất bản" với mục đích sử dụng lâu dài cho nhiều subscribers khác nhau, mỗi subscriber có nhu cầu và vòng đời riêng. Kafka lưu trữ tin nhắn trong topic để subscribers có thể truy cập lại sau.

Câu hỏi 8: Sự khác nhau giữa phân tán dọc và phân tán ngang là gì?

|  |  |  |
| --- | --- | --- |
|  | Phân tán dọc | Phân tán ngang |
| Cách thức tổ chức | * Dữ liệu và xử lý được phân chia theo chức năng hoặc lớp (layer). * Mỗi lớp hoặc thành phần chịu trách nhiệm cho một nhiệm vụ cụ thể. | * Dữ liệu và xử lý được phân chia theo các node hoặc máy chủ độc lập. * Mỗi node có thể thực hiện cùng một nhiệm vụ hoặc xử lý một phần dữ liệu. |
| **Ưu điểm** | * Dễ quản lý và bảo trì do các lớp được tách biệt rõ ràng. * Tối ưu hóa hiệu suất bằng cách chuyên môn hóa từng lớp. | * Khả năng mở rộng cao: Có thể thêm nhiều node để xử lý tải công việc lớn hơn. * Tăng tính sẵn sàng và khả năng chịu lỗi: Nếu một node gặp sự cố, các node khác vẫn hoạt động. |
| **Nhược điểm** | * Khó mở rộng: Việc thêm nhiều lớp hoặc chức năng có thể làm tăng độ phức tạp. * Phụ thuộc giữa các lớp: Lỗi ở một lớp có thể ảnh hưởng đến toàn bộ hệ thống. | * Khó quản lý dữ liệu nhất quán: Cần các cơ chế đồng bộ hóa phức tạp. * Chi phí cao: Cần nhiều tài nguyên để duy trì và quản lý các node. |

**Câu hỏi 9:** Phân tích ưu nhược điểm của kiến trúc tập trung và kiến trúc không tập trung.

|  |  |  |
| --- | --- | --- |
|  | Kiến trúc tập trung | Kiến trúc không tập trung |
| Ưu điểm | * **Dễ quản lý:** Tất cả được tập trung ở một nơi, giúp quản lý và giám sát đơn giản hơn. * **Hiệu suất cao:** Máy chủ trung tâm xử lý nhanh hơn do không cần phối hợp nhiều nút, phù hợp cho các tác vụ cần phản hồi nhanh. * **Tiết kiệm chi phí:** Không cần đầu tư vào nhiều máy chủ, giảm chi phí thiết lập và bảo trì. * **Tính nhất quán dữ liệu:** Dữ liệu được lưu trữ ở một nơi, dễ đảm bảo không có sự không đồng nhất. | * **Độ tin cậy cao:** Không có điểm lỗi duy nhất, hệ thống vẫn hoạt động nếu một số nút gặp sự cố. * **Dễ mở rộng:** Có thể thêm nút để xử lý tải tăng mà không gây quá tải trung tâm. * **Bảo mật tốt hơn:** Dữ liệu phân tán, khó tấn công toàn bộ hệ thống; dữ liệu có thể được mã hóa và sao chép trên nhiều nút. * **Tính tự chủ:** Mỗi nút hoạt động độc lập, tăng tính linh hoạt. |
| Nhược điểm | * **Điểm lỗi duy nhất:** Nếu máy chủ trung tâm gặp sự cố, toàn bộ hệ thống sẽ ngừng hoạt động. * **Khó mở rộng:** Khi hệ thống lớn lên, máy chủ trung tâm có thể không xử lý được tải tăng, dẫn đến hiệu suất giảm. * **Rủi ro bảo mật:** Máy chủ trung tâm là mục tiêu chính cho hacker; nếu bị tấn công, toàn bộ hệ thống bị ảnh hưởng. | * **Phức tạp hơn:** Thiết kế và quản lý phức tạp do cần phối hợp giữa nhiều nút. * **Hiệu suất có thể chậm:** Độ trễ cao hơn do cần giao tiếp giữa các nút, đặc biệt khi dữ liệu ở xa. * **Khó duy trì tính nhất quán dữ liệu:** Đảm bảo dữ liệu đồng nhất trên tất cả nút là thách thức, đòi hỏi cơ chế đồng bộ phức tạp. * **Chi phí cao hơn:** Thiết lập và bảo trì nhiều nút tốn kém hơn so với hệ thống tập trung. |

**Câu hỏi 10:** Trong một mạng overlay có cấu trúc, các thông điệp được định tuyến dựa theo hình trạng mạng (topology). Nhược điểm quan trọng của hướng tiếp cận này là gì?

Trong mạng **overlay có cấu trúc** (structured overlay network), các nút được tổ chức theo một hình trạng xác định (ví dụ: cây, vòng, lưới, hoặc DHT - Distributed Hash Table), và việc định tuyến thông điệp dựa trên logic của cấu trúc này.

**Độ phức tạp và chi phí duy trì cấu trúc**

* **Vấn đề**: Cấu trúc overlay (ví dụ: DHT) yêu cầu các nút phải liên tục cập nhật thông tin về trạng thái mạng (ví dụ: danh sách nút lân cận, khóa phân tán). Khi có sự thay đổi (nút gia nhập/rời khỏi mạng), hệ thống phải tái cân bằng cấu trúc, gây tốn tài nguyên tính toán và băng thông.

**Cân bằng tải không đồng đều**

* **Vấn đề**: Việc phân phối khóa trong cấu trúc overlay (ví dụ: DHT) có thể dẫn đến tải không đồng đều giữa các nút. Một số nút có thể phải lưu trữ hoặc xử lý nhiều dữ liệu hơn các nút khác

**Khó triển khai trong mạng không đồng nhất**

* **Vấn đề**: Cấu trúc overlay giả định các nút có khả năng xử lý và băng thông tương đương. Trong thực tế, mạng thường không đồng nhất (heterogeneous), với các nút có tài nguyên khác nhau (ví dụ: điện thoại, máy chủ). Điều này gây khó khăn cho việc tối ưu hóa hiệu năng.

**Câu hỏi 11:** Xét một chuỗi các tiến trình P1, P2, ..., Pn triển khai một kiến trúc client-server đa tầng. Cơ chế hoạt động của tổ chức đó như sau: tiến trình Pi là client của tiến trình Pi+1, và Pi sẽ trả lời Pi-1 chỉ khi đã nhận được câu trả lời từ Pi+1.

Vậy những vấn đề nào sẽ nảy sinh với tổ chức này khi xem xét hiệu năng yêu cầu-trả lời tới P1?

Với mô hình kiến trúc phân tầng như trên để từ p1 gửi đến tiến trình pn cần n-1 ần trao đổi qua lại. Như vậy, khi 1 giai đoạn nào đó gặp vấn đề về timeout hay vì một lý do nào đó cần gửi lại yêu cầu thì thời gian phản hồi sẽ rất lâu. Hơn nữa việc trao đổi qua lại quá nhiều lần làm tỉ lệ lỗi truyền tải càng cao hơn.

**Câu hỏi 12:** Xét mạng CAN như trong hình. Giả sử tất cả các node đều biết node hàng xóm của mình. Một giải thuật định tuyến được đưa ra đó là gửi các gói tin cho node hàng xóm gần mình nhất và hướng đến đích. Giải thuật này có tốt không? Giải thích.

Ta có thể tính được đường đi ngắn nhất từ (0.2, 0.3) đến (0.9, 0.9) là qua (0.6, 0.7), tuy nhiên theo giải thuật CAN thì gửi gói tin cho node hàng xóm gần nhất trước nên tuyến đường sẽ đi qua (0.2, 0.8). Vì vậy có thể thấy giải thuật này chưa phải giải thuật tốt nhất. Tổng khoảng cách từ đầu tới cuối tối ưu nhất chưa chắc đã đi qua những điểm gần nó nhất.