**BT MÔN HỆ THỐNG PHÂN TÁN**

**Nhóm 4:**

B24CHHT085 - Nguyễn Khắc Minh

B24CHHT074 - Phạm Văn Hoàng

B24CHHT071 - Nguyễn Văn Hiếu

**Câu 1:**

- Hai đặc điểm quan trọng của hệ thống phân tán:

* đặc điểm thứ nhất đó là tập hợp các phần tử tính toán có thể hoạt động độc lập với nhau
* đặc điểm thứ hai là chúng cần phải cộng tác để giải quyết một nhiệm vụ chung.

Ba lý do cơ bản khiến các ứng dụng phân tán phức tạp hơn so với các ứng dụng đơn lẻ. Các ứng dụng phân tán phức tạp hơn so với các ứng dụng đơn lẻ vì chúng phải đối mặt với những vấn đề không tồn tại trong hệ thống đơn lẻ, bao gồm:

* **Độ tin cậy của mạng:** Mạng không đáng tin cậy đòi hỏi phải giải quyết vấn đề trong suốt về lỗi truyền thông. Trong các hệ thống phân tán, mạng không thể được coi là luôn đáng tin cậy, an toàn, đồng nhất, có hình trạng không đổi, độ trễ bằng không, băng thông vô hạn, chi phí vận chuyển bằng không, và chỉ có một người quản trị. Những giả định này có thể đúng với ứng dụng độc lập nhưng lại là vấn đề lớn trong hệ thống phân tán.
* **Vấn đề hiệu năng:** Trong nhiều trường hợp, tính toán phân tán bắt buộc phải sử dụng để tận dụng khả năng tính toán song song hoặc khai thác khả năng của các máy tính chuyên dụng.
* **Khả năng chịu lỗi:** Hệ thống cần đảm bảo an toàn tuyệt đối ngay cả khi có sự cố trên một máy chủ nào đó mà không ảnh hưởng đến hoạt động toàn bộ hệ thống. Điều này đòi hỏi phải phát hiện kịp thời và xử lý lỗi, nếu máy chủ hỏng hoàn toàn thì chuyển sang máy tính dự phòng khác để dịch vụ không bị gián đoạn.

**Câu 2:**

- Hiệu năng hệ thống sẽ bị hạn chế bởi băng thông của kênh truyền và hầu hết các đơn vị xử lý trung tâm sẽ lãng phí thời gian chờ để đến lượt đọc/ghi ô nhớ. Nếu chỉ có hai hoặc ba đơn vị xử lý trung tâm thì việc quản lý tương tranh khá đơn giản, vấn đề sẽ trở nên khá phức tạp khi số lượng của chúng tăng lên. Để giải quyết vấn đề này, người ta xây dựng kiến trúc truy nhập bộ nhớ không đồng nhất bằng cách thêm vùng đệm vào bo mạch của bộ vi xử lý. Vùng đệm trao đổi thông tin với bộ nhớ dùng chung theo phương pháp đọc/ghi từng khối 32 hoặc 64 byte, đơn vị xử lý trung tâm sẽ đọc/ghi các ô nhớ trong vùng đệm, như vậy sẽ tăng hiệu năng sử dụng kênh truyền.

Để mở rộng vùng đệm, có thể thêm bộ nhớ cục bộ được truy nhập bằng kênh riêng cho mỗi bộ vi xử lý, như vậy cần giải quyết vấn đề nhân bản ô nhớ. Mỗi khối ô nhớ được đánh dấu trạng thái đọc hoặc ghi, nếu đơn vị xử lý trung tâm muốn ghi một ô nhớ được nhân bản ở bộ nhớ riêng thì phần cứng của kênh truyền sẽ gửi tín hiệu ghi đến các bộ nhớ này. Trước khi thay đổi ô nhớ trong vùng đệm thì phải lưu lại giá trị cũ vào bộ nhớ dùng chung hoặc chuyển trực tiếp giá trị đó đến bộ điều khiển ghi qua kênh truyền. Để sử dụng một cách tối ưu, bộ nhớ riêng sẽ lưu mã chương trình, các hằng số, dữ liệu chỉ đọc và các biến số riêng, bộ nhớ dùng chung sẽ được cấp phát cho các biến số khác.

Công nghệ này cho phép người dùng bắt đầu với những máy chủ tương đối nhỏ và sau đó bổ sung bộ vi xử lý nếu cần mở rộng. Nếu sử dụng kiến trúc truy nhập bộ nhớ đồng nhất thì việc bổ sung thêm bộ vi xử lý không làm tăng hiệu năng như kỳ vọng, kiến trúc truy nhập bộ nhớ không đồng nhất khắc phục nhược điểm này và cho phép mở rộng lên đến 256 bộ vi xử lý trên một máy. Thông thường các bộ xử lý được phân thành những nhóm nhỏ, trong đó tất cả các bộ vi xử lý liên kết với nhau và có bộ nhớ riêng cho mỗi nhóm, nó làm giảm tình trạng tắc nghẽn vì các bộ vi xử lý trong một nhớm giao tiếp với nhau và với bộ nhớ cục bộ của chúng qua những kênh riêng.

- Hệ điều hành là một phần mềm đặc biệt dùng để điều hành và quản lý toàn bộ tất cả thành phần bao gồm cả phần cứng lẫn phần mềm ứng dụng, nó đóng vai trò nền tảng cung cấp các dịch vụ truy nhập tài nguyên. Hệ điều hành đơn giản hóa sự phức tạp và đa dạng của phần cứng bằng kỹ thuật ảo hóa, như vậy nếu phần cứng được nâng cấp hoặc sản phẩm của các hãng khác nhau thì không cần phải thay đổi phần mềm ứng dụng. Hệ điều hành được thiết kế để phù hợp với nhiều chủng loại thiết bị khác nhau, từ máy tính cho đến điện thoại thông minh hoặc các thiết bị trung gian như thiết bị định tuyến và thiết bị chuyển mạch. Hệ điều hành mạng thường dùng cho các hệ thống không đồng nhất, nó cung cấp các tính năng để có thể kết nối mạng, mỗi máy tính tạo ra các dịch vụ cung cấp cho các máy tính khác. Ngoài các chức năng cơ bản của một hệ điều hành, các hệ điều hành mạng còn phải thực hiện việc chia sẻ và bảo vệ tài nguyên của mạng. Các hệ điều hành Microsoft Server, Linux, Unix được xếp vào loại hệ điều hành mạng, máy chủ quản lý tài nguyên của mình để cung cấp dịch vụ cho các máy khách. Quan hệ giữa các máy tính không chặt chẽ và sự gắn kết trong hệ thống do các ứng dụng phân tán đảm nhiệm, một số các dịch vụ chung do hệ điều hành cung cấp cho các ứng dụng phân tán gọi là phần mềm trung gian. Phần mềm trung gian thường được triển khai dưới dạng thư viện, những thư viện này cung cấp các phương tiện trao đổi thông tin dựa trên thông điệp.

Nếu máy tính chỉ lắp đặt một bộ vi xử lý thì nhiệm vụ của hệ điều hành chỉ đơn giản là quản lý tiến trình và ảo hóa tài nguyên để các phần mềm ứng dụng dễ dàng truy nhập. Các tiến trình sử dụng tài nguyên chung nhưng vẫn đảm bảo tính độc lập, do đó hệ điều hành cần phải giải quyết vấn đề tương tranh và ảo hóa tài là cách tiếp cận đơn giản nhất, nó cung cấp khẳ năng xử lý đa nhiệm cho các ứng dụng. Bên trong hệ điều hành thường có phần lõi, nó quản lý phần cứng, lập lịch cho các tiến trình và xử lý các yêu cầu

tương tác giữa phần cứng và phần mềm, nghĩa là có thể truy nhập trực tiếp vào bộ nhớ và các thanh ghi. Dựa trên các hàm do phần lõi cung cấp sẽ phát triển thêm các hàm giao diện lập trình để hạn chế truy nhập trực tiếp vào các thanh ghi hoặc bộ nhớ.

Phần mềm ứng dụng thường truy nhập đến tài nguyên thông qua các hàm do hệ điều hành cung cấp, chúng có thể thuộc loại phong tỏa không phong tỏa. Hàm phong tỏa nghĩa là chương trình gọi sẽ phải chờ kết quả thực hiện, nó không cho phép xử lý song song, như vậy toàn bộ tiến trình cũng sẽ bị phong tỏa. Hàm không phong tỏa cho phép chương trình gọi tiếp tục thực hiện mà không cần chờ đợi lời gọi hoàn thành, đây là điểm quan trọng để có thể áp dụng kỹ thuật lập trình song song nhằm nâng cao hiệu năng xử lý.

Đối với hệ thống gồm nhiều bộ vi xử lý, hệ điều hành có thể được thiết kế đối xứng hoặc bất đối xứng. Nếu thiết kế đối xứng thì mỗi tiến trình có thể chạy trên bất kỳ bộ vi xử lý nào, các bộ vi xử lý trao đổi thông tin qua một bộ nhớ dùng chung. Ưu điểm của nó là tính chịu lỗi và khả năng cân bằng tải tối ưu hơn, vì các tiến trình của hệ điều hành có thể chạy trên bất kỳ bộ xử lý nào nên nguy cơ tắc nghẽn giảm đi đáng kể. Tuy nhiên vấn đề đồng bộ giữa các bộ vi xử lý được đặt lên hàng đầu khi thiết kế hệ điều hành cho hệ thống đối xứng. Trong thiết kế bất đối xứng, hệ điều hành dành ra một hoặc hai bộ xử lý để sử dụng riêng, các bộ xử lý còn lại dùng để điều khiển các chương trình của người sử dụng. Hệ thống bất đối xứng đơn giản hơn nhiều so với hệ thống đối xứng, nhưng nếu bộ vi xử lý dành riêng cho hệ điều hành bị hỏng thì hệ thống có thể ngừng hoạt động.

**Câu 3:**

Một trong những vấn đề cơ bản của hệ thống phân tán là phải đảm bảo vấn đề hiệu năng, tính toán phân tán thường được sử dụng trong các tác nghiệp yêu cầu hiệu năng cao. Tùy theo nền tảng hệ thống, điện toán phân tán có thể triển khai theo hình thức cụm hoặc lưới, dù theo hình thức nào thì điểm cốt lõi của nó vẫn là việc phân chia nhiệm vụ để xử lý trên nhiều máy tính. Trong hệ thống điện toán cụm, nền tảng phần cứng là tập hợp các máy tính tương tự nhau, chúng được cài đặt các hệ điều hành giống nhau và kết nối với nhau qua đường truyền tốc độ cao, thông thường là mạng cục bộ. Ngược lại, hệ thống điện toán lưới bao gồm các hệ thống phân tán thuộc về nhiều miền quản lý khác nhau và thường không đồng nhất về phần cứng, phần mềm, hệ điều hành và các hệ thống này có thể sử dụng những công nghệ mạng khác nhau.

***Hệ thống điện toán cụm:***

Giá thành máy tính ngày càng giảm trong khi khả năng xử lý ngày càng tăng là cơ hội để xây dựng hệ thống điện toán cụm, các máy tính được kết nối với nhau trong mạng tốc độ cao. Nguyên lý của nó vẫn là sử dụng kỹ thuật xử lý song song trên nhiều máy tính để tăng hiệu năng xử lý, người sử dụng nhìn nhận điện toán cụm như một máy tính ảo với hiệu năng cao. Trên mỗi máy tính có thể sử dụng hệ điều hành phân tán để liên kết các máy tính, nhưng cũng có thể chỉ cần cài đặt hệ điều hành mạng và chạy phần mềm hỗ trợ điện toán cụm.

Một ví dụ khá quen biết của điện toán cụm là hệ thống Beowulf chạy trên hệ điều hành Linux, nó được thiết kế bất đối xứng, nghĩa là tồn tại nút chủ đảm nhiệm chức năng xắp đặt vị trí của các thành viên khác trong chương trình song song, quản lý hàng đợi các công việc và giao tiếp với người sử dụng. Hình 1.2 thể hiện loại hệ điều hành này, nút chủ chỉ việc chạy phần mềm trung gian cần thiết cho các chương trình thực hiện và quản lý cụm, trong khi đó các nút tính toán sẽ không cần gì khác ngoài hệ điều hành chuẩn.

Một ví dụ khác là hệ thống MOSIX được xây dựng dựa trên cách tiếp cận đối xứng, nghĩa là không có máy tính nào đóng vai trò nút chủ. Độ trong suốt rất cao của MOSIX đạt được bằng cách di trú tiến trình, người sử dụng có thể khởi tạo tiến trình trên một máy tính nhưng tiến trình đó có thể di trú sang máy tính khác để thực hiện nhằm tận dụng tối đa tiềm năng xử lý của mỗi nút trong hệ thống.

Dù sử dụng cách nào thì điều kiện tiên quyết để có thể triển khai hệ thống điện toán cụm là nền tảng phải đồng nhất với kênh truyền tốc độ cao, nền tảng ở đây bao gồm phần cứng cũng như hệ điều hành và phần mềm hỗ trợ. Điện toán cụm thường áp dụng cho các trung tâm tính toán lớn, các máy tính kết nối với nhau trong mạng cục bộ đảm bảo kênh truyền tốc độ cao và ổn định.

***Hệ thống điện toán lưới:***

Không phải lúc nào cũng đảm bảo tính đồng nhất về hạ tầng để cài đặt hệ thống điện toán cụm, nếu nhìn xa hơn sẽ thấy điện toán toán cụm vẫn rơi vào cách xử lý tập trung, dù cho hiệu năng của nó rất cao nhưng vẫn không tránh khỏi việc tập trung các yêu cầu về một địa điểm, điều này có thể dẫn tới nghẽn mạng. Hơn nữa, việc giải quyết các yêu cầu của người sử dụng có thể liên quan đến nhiều đơn vị khác nhau, không thể bắt buộc máy tính của các đơn vị này cài đặt cùng một loại hệ điều hành và kết nối với nhau qua kênh truyền tốc độ cao, đó là chưa tính đến mỗi cơ quan lại có những chính sách quản lý riêng.

Điện toán lưới là mô hình tính toán dựa trên mạng để có khả năng xử lý một lượng lớn dữ liệu, một nhóm các máy tính phối hợp với nhau để giải quyết vấn đề chung. Tương tự như điện toán cụm, các máy tính trong hệ thống điện toán lưới liên kết với nhau để giải quyết một vấn đề chung bằng cách chia thành các đơn vị nhỏ hơn để xử lý, chúng hình thành một siêu máy tính ảo. Hệ thống điện toán lưới không đòi hỏi tính đồng nhất của tất cả các nút, mỗi thành viên có thể khác về cả phần cứng lẫn hệ điều hành và các chính sách quản lý.

Vấn đề cốt lõi trong hệ thống điện toán lưới là việc quản lý tài nguyên của các cơ quan khác nhau nhưng phải cho phép các nhóm người sử dụng thuộc các cơ quan cộng tác với nhau, như vậy sự cộng tác được thực hiện dựa trên cơ quan ảo, người sử dụng thuộc về một cơ quan ảo thì có quyền truy nhập đến các tài nguyên của cơ quan ảo đó. Với đặc tính đó, nhiều phần mềm hệ thống điện toán lưới phát triển xung quanh việc truy nhập tài nguyên từ các vùng quản trị khác nhau cho người sử dụng và ứng dụng thuộc về một cơ quan ảo, do đó tiêu điểm của hệ thống điện toán lưới thường là những vấn đề liên quan đến mô hình hệ thống.

Tầng ứng dụng bao gồm các ứng dụng vận hành bên trong cơ quan ảo và sử dụng môi trường điện toán lưới. Tầng tiếp nhận xử lý các yêu cầu truy nhập đến nhiều tài nguyên khác nhau, thường cung cấp các chức năng như: thăm dò, định vị, lập lịch truy nhập , nhân bản tài nguyên... Các giao thức thuộc tầng này khá nhiều, để đảm bảo cung cấp dịch vụ theo yêu cầu của tầng ứng dụng nên chúng thường không phải là những giao thức đã được chuẩn hóa.

Tầng kết nối bao gồm các giao thức truyền thông để hỗ trợ cho các giao tác lưới bao trùm toàn bộ các tài nguyên, ví dụ các giao thức truy nhập để di chuyển tài nguyên hoặc đơn giản chỉ là truy nhập tài nguyên từ một vị trí nào đó. Tầng kết nối sẽ phải bao gồm các các giao thức bảo mật, tính năng bảo mật có thể cho một tài khoản và cũng có thể cho một ứng dụng.

Tầng tài nguyên quản lý tài nguyên đơn lẻ, nó sử dụng các chức năng do tầng kết nối cung cấp và gọi trực tiếp các giao diện tầng kết cấu cung cấp để thực hiện các chức năng điều khiển truy nhập, ví dụ các chức năng thiết lập cấu hình tài nguyên, khởi tạo tiến trình đọc/ghi dữ liệu. Tầng thực hiện cung cấp giao diện để truy nhập tài nguyên cục bộ, các giao diện này được xây dựng để có thể thích ứng với việc cho phép chia sẻ tài nguyên bên trong một cơ quan ảo, nó thường cung cấp các chức năng để truy vấn trạng thái và khả năng của tài nguyên, các chức năng quản lý tài nguyên thực.

Trong các hệ thống điện toán lưới, tầng tiếp nhận, tầng kết nối và tầng tài nguyên thường được gộp lại vào tầng trung gian, nó có nhiệm vụ quản lý và cung cấp chức năng truy nhập trong suốt đến tất cả các tài nguyên phân bố trên các trang mạng khác nhau. Quan sát cho thấy, việc cung cấp các thông tin riêng lẻ trong các hệ thống điện toán lưới khá phổ biến, điều này đã dần dẫn tới quan điểm về mô hình dịch vụ lưới mở.

***Hệ thống thông tin phân tán:***

Thông tin phân tán là thực tế khá phổ biến trong hệ thống phân tán, dữ liệu được lưu trữ trên nhiều máy tính, xử lý dữ liệu trên một máy có thể sẽ liên quan đến những máy khác. Việc dữ liệu được lưu trữ phân tán xuất phát từ nhiều nguyên nhân, có thể do dung lượng dữ liệu lớn khi qui mô phát triển hoặc triển khai kỹ thuật nhân bản để tăng hiệu năng và đảm bảo tính tin cậy. Các phần mềm ngày nay thường được xây dựng theo

mô hình ba bên, tách biệt các thành phần lưu trữ dữ liệu với xử lý nghiệp vụ và tiếp nhận yêu cầu.

Nếu dữ liệu được lưu trữ tập trung, khi nhận được yêu cầu từ máy khách thì máy chủ dịch vụ chỉ cần gửi yêu cầu truy vấn đến một máy chủ cơ sở dữ liệu, kết quả nhận được sẽ xử lý và trả về cho máy khách. Vấn đề sẽ phức tạp hơn nếu dữ liệu được nhân bản, các yêu cầu đọc sẽ không có nhiều điểm khác biệt, dù cho đôi khi phải thực hiện đọc phân tán, nhưng các yêu cầu ghi lại khác hoàn toàn, thay đổi dữ liệu phải được thực hiện thống nhất trên tất cả các bản sao.

Nếu máy khách truy nhập đến từng ứng dụng trên máy chủ để thực hiện trên các thao tác dữ liệu như các hệ thống hỏi/đáp thì sẽ rất phức tạp, thời gian thực hiện kéo dài và thậm chí khó có thể đạt được mục tiêu nhất quán. Hình 1.4 thể hiện cách truy nhập dữ liệu trong các hệ thống thông tin phân tán, đưa các yêu cầu truy vấn vào tầng trung gian sẽ đơn giản hóa cách phát triển hệ thống, tầng trung gian sẽ quyết định gửi yêu cầu đến những máy chủ nào và tổng hợp kết quả trả về cho máy khách. Nếu là yêu cầu đọc, nó có thể quyết định xem có cần đọc phân tán hay không, kỹ thuật lưu trữ cache cũng có thể được áp dụng để nâng cao hiệu năng hệ thồng và giảm thời gian phản hồi kết quả. Nếu là yêu cầu ghi thì phần mềm trung gian sẽ lựa chọn mô hình nhân bản để cân bằng yêu cầu của người sử dụng và các vấn đề liên quan đến hiệu năng hệ thống.

***Hệ thống lan tỏa phân tán:***

Các máy chủ thường được đặt ở vị trí địa lý cố định và kết nối với nhau qua kênh truyền chất lượng cao, tính ổn định này tạo điều kiện thuận lợi để đạt được tính trong suốt trong xử lý cũng như truy nhập tài nguyên. Các ký thuật xử lý phân tán và nhân bản dữ liệu không những nâng cao hiệu năng hệ thống mà còn có thể che giấu lỗi, nếu xảy ra trên một máy chủ nào đó. Vấn đề còn lại là các máy khách kết nối đến hệ thống máy chủ như thế nào và cách truyền dữ liệu ra sao.

Vấn đề trở nên phức tạp khi xuất hiện nhiều chủng loại thiết bị trong các gia đình, thậm chí các thiết bị di động ngày càng phổ biến, chúng thường được kết qua môi trường

không dây với đặc trưng là tính không ổn định về vị trí cũng như tốc độ truyền dẫn. Hệ thống này thiếu vắng sự kiểm soát nhân công, cấu hình của các thiết bị do chủ sở hữu qui định hoặc thiết bị sẽ tự động khám phá môi trường và lựa chọn cấu hình phù hợp nhất. Môi trường như vậy gọi là lan tỏa phân tán, để tạo điều kiện lựa chọn cấu hình chính xác nhất, Grimm đã đề xuất các thiết bị phải có khả năng bao quát ngữ cảnh, cung cấp giao diện cấu hình mặc định và tự động nhận biết chia sẻ tài nguyên.

Bao quát thay đổi ngữ cảnh nghĩa là thiết bị phải liên tục nhận biết được môi trường có thể thay đổi bất kỳ thời gian nào, ví dụ khi phát hiện thấy mất kết nối mạng thì thiết bị sẽ tự động tìm mạng khác thay thế. Cung cấp giao diện cấu hình mặc định nghĩa là mỗi người sử dụng có thói quen riêng biệt, do đó cần phải cung cấp giao diện cấu hình sao cho đơn giản nhất phù hợp với tất cả mọi người hoặc một cấu hình được cài đặt tự động.

Tự động nhận biết chia sẻ tài nguyên là đặc điểm quan trọng của hệ thống lan tỏa, các thiết bị tham gia hệ thống theo thứ tự truy nhập thông tin, điều này đòi hỏi phải cung cấp các phương tiện để dễ dàng đọc, lưu trữ, quản lý và chia sẻ thông tin. Với quan niệm việc kết nối mạng của các thiết bị thường gián đoạn hoặc thay đổi thì không gian lưu trữ thông tin có thể truy nhập được cũng sẽ phải thay đổi theo thời gian. Với sự hiện diện của khả năng di động thì các thiết bị phải dễ dàng thích nghi với môi trường cục bộ, chúng phải có khả năng dễ dàng phát hiện các dịch vụ và phản hồi theo các dịch vụ đó. Sự trong suốt về mặt phân bố không chỉ ở trong hệ thống lan tỏa mà thực tế còn ở sự phân bố về dữ liệu, xử lý và điều khiển, vì lẽ đó tốt hơn hết là phải phơi bày chứ không nên che giấu chúng.

Cùng với xu thế Internet vạn vật, hệ thống lan tỏa phát triển ngày càng mạnh mẽ, ví dụ hệ thống thiết bị trong gia dụng, hệ thống chăm sóc sức khỏe, mạng cảm biến..., khó có thể yêu cầu người dùng cài đặt phần mềm trên những thiết bị này, chúng phải được thiết kế sao cho có thể tự động nhận biết được môi trường xung quanh. Lượng dữ liệu thu được từ các thiết bị này rất lớn, giải pháp lắp đặt cho mỗi thiết bị một bộ lưu trữ lên tới vài TB là không khả thi, cho dù có giải quyết được vấn đề lưu trữ thì vẫn gặp phải một loạt những vấn đề khác, tùy từng hệ thống cần phải xem xét các vấn đề sau:

- Bảo mật tính riêng tư của thông tin.

- Dữ liệu được lưu trữ ở đâu và bằng cách nào có thể khai thác được chúng.

- Các biện pháp phòng chống mất mát dữ liệu quan trọng.

- Hạ tầng cần thiết để tạo và lan tỏa các cảnh báo.

- Phương pháp phản hồi và kiểm soát trực tuyến các thiết bị.

- Sử dụng giải pháp nào để triển khai một lượng rất lớn các thiết bị.

- Giải pháp khắc phục khi gặp sự cố về đường truyền mạng.

Đối với các thiết bị gia dụng, vấn đề ưu tiên hàng đầu là các thiết bị phải có khả năng tự cấu hình quản lý và tính riêng tư của thông tin, ví dụ hệ thống chăm sóc sức khỏe lại đặt vấn đề cảnh báo và phản hồi trực tuyến là quan trọng, trong khi đó hệ thống cảm biến lại phải đương đầu với số lượng rất lớn các thiết bị. Lưu trữ dữ liệu cục bộ tạo điều kiện cho thiết bị hoạt động tương đối độc lập với môi trường mạng, tuy nhiên vấn đề khai

thác dữ liệu sẽ trở nên phức tạp, hơn nữa nó đòi hỏi không gian lưu trữ đủ lớn trên mỗi thiết bị.

**Câu 4:**

- Một trong những mục tiêu quan trọng khi thiết kế hệ thống sao cho nó có khả năng tự phục hồi khi gặp lỗi và không ảnh hưởng nghiêm trọng đến hiệu năng tổng thể. Bất cứ khi nào lỗi xảy ra, hệ thống vẫn tiếp tục hoạt động theo cách chấp nhận được trong thời gian sửa lỗi, nghĩa là hệ thống có khả năng chịu lỗi".

**Ba hình thức “tính trong suốt” (trong suốt về truy nhập, vị trí và lỗi), và ví dụ đơn giản minh họa mỗi loại:**

* **Trong suốt về truy nhập (Access Transparency):** "Tính trong suốt truy nhập tài nguyên nhân bản". Người dùng hoặc ứng dụng có thể truy cập tài nguyên mà không cần biết liệu tài nguyên đó có phải là bản sao hay không, hoặc cần xử lý cụ thể cho bản sao đó.
* **Trong suốt về vị trí (Location Transparency**: Hệ thống MOSIX đạt được độ trong suốt rất cao bằng cách di trú tiến trình. Người sử dụng có thể khởi tạo tiến trình trên một máy tính nhưng tiến trình đó có thể di trú sang máy tính khác để thực hiện nhằm tận dụng tối đa tiềm năng xử lý của mỗi nút trong hệ thống. Điều này cho thấy người dùng không cần biết vị trí vật lý của tiến trình đang chạy.
* **Trong suốt về lỗi (Failure Transparency):** Mạng không đáng tin cậy đòi hỏi phải giải quyết vấn đề trong suốt về lỗi truyền thông. Điều này có nghĩa là hệ thống nên che giấu các lỗi xảy ra (ví dụ: lỗi mạng, lỗi máy chủ) để người dùng hoặc các thành phần khác không bị ảnh hưởng trực tiếp bởi lỗi đó và hệ thống vẫn tiếp tục hoạt động.

**Mối quan hệ giữa “tính mở” (openness) và khả năng tương tác (interoperability) trong hệ thống phân tán:**

-Hệ thống phân tán là khả năng một hệ thống có thể kết nối với các hệ thống khác, hoặc dễ dàng bổ sung, thay thế các thành phần của nó. Để đảm bảo tính mở, các thành phần của hệ thống phân tán phải được thiết kế dựa trên các giao diện chuẩn và phải có khả năng tương tác với nhau.

Mối quan hệ giữa tính mở và khả năng tương tác được thể hiện qua các điểm sau:

* **Tiêu chuẩn hóa:** Để một hệ thống có tính mở cao, các thành phần của nó phải tuân thủ các giao diện chuẩn và mô tả chính xác cách tương tác với nhau. Điều này trực tiếp liên quan đến khả năng tương tác.
* **Khó khăn với hệ thống "khối":** Nhiều hệ thống cũ và một số hệ thống hiện nay được xây dựng dựa trên cách tiếp cận "khối", các thành phần tách biệt về logic nhưng được biên dịch trong một chương trình lớn, khó thay thế hoặc thích nghi mà không ảnh hưởng đến toàn bộ hệ thống. Các hệ thống như vậy có xu thế đóng chứ không có tính mở. Điều này cho thấy thiếu khả năng tương tác giữa các thành phần làm giảm tính mở.
* **Tách biệt chính sách và cơ chế:** Để hệ thống có tính mở, cần tách biệt giữa chính sách với các cơ chế xử lý, tổng quát hóa và cụ thể hóa bằng cách cung cấp các tham số để người sử dụng có thể tùy biến. Điều này tạo điều kiện cho các thành phần khác nhau tương tác linh hoạt hơn.

**Câu 5:**

**- Kiến trúc phân cấp (Client-Server Architecture):**

* + **Ưu điểm:** Mô hình khách/chủ và vai trò quan trọng của nó trong hệ thống phân tán. Trong mô hình này, máy khách chứa chương trình cài đặt chức năng tầng giao diện, còn máy chủ thực hiện các chức năng tầng nghiệp vụ và tầng dữ liệu. Điều này giúp tập trung quản lý và bảo mật dữ liệu.
  + **Nhược điểm:** Nếu máy chủ thực hiện mọi công việc thì máy khách vẫn chỉ là đầu cuối câm. Việc chuyển các chức năng của tầng xử lý nghiệp vụ hoặc một phần chức năng của tầng dữ liệu về máy khách không phải là giải pháp tốt vì nguy cơ lộ thông tin truy nhập cơ sở dữ liệu và làm hệ thống phụ thuộc nhiều hơn vào máy khách.

**-Kiến trúc ngang hàng (Peer-to-Peer Architecture):**

* + **Ưu điểm:** Kiến trúc ngang hàng và trong tổ chức nhóm ngang hàng, tất cả các tiến trình đều có vai trò như nhau và quyết định mang tính tập thể. Khi có yêu cầu gửi đến, tất cả các thành viên đều nhận được. Nếu một tiến trình bị lỗi thì chỉ làm giảm kích thước của nhóm. Điều này cho thấy tính chịu lỗi và phân tán tốt hơn.
  + **Nhược điểm:** Việc "chỉ một thành viên được phép xử lý yêu cầu, nếu không sẽ dẫn đến lỗi lặp, do đó các tiến trình phải thực hiện giải thuật bầu chọn", điều này có thể làm tăng độ phức tạp trong việc quản lý và đồng bộ.

**-Bốn mô hình hệ thống phân tán (phân tầng, đối tượng phân tán, kênh sự kiện, dữ liệu tập trung) và ví dụ ứng dụng điển hình cho mỗi mô hình:**

* **Mô hình phân tầng (Layered Model):**
  + **Mô tả:** Nhiều hệ thống phân tán sử dụng mô hình phân tầng, tách biệt giữa các chức năng tiếp nhận yêu cầu với xử lý nghiệp vụ và lưu trữ dữ liệu. Tương tác giữa các tầng thường sử dụng mô hình khách/chủ.
  + **Ví dụ ứng dụng:** các ứng dụng khách/chủ nơi chức năng được phân chia thành các tầng rõ ràng (giao diện, nghiệp vụ, dữ liệu).
* **Mô hình đối tượng phân tán (Distributed Object Model):**
  + **Mô tả:** Ràng buộc lỏng hơn mô hình phân tầng. Mỗi đối tượng được coi là một thành phần và được kết nối thông qua cơ chế gọi thủ tục từ xa (RPC). Các đối tượng được gọi trực tiếp nên thời gian trễ thấp và phù hợp với ứng dụng yêu cầu thời gian thực.
  + **Ví dụ ứng dụng:** Phù hợp với những ứng dụng yêu cầu thời gian thực.
* **Mô hình kênh sự kiện (Event Channel Model):**
  + **Mô tả:** Kênh sự kiện là một tiến trình trung gian cho các tiến trình khác trong hệ thống. Mỗi thành phần cung cấp dịch vụ sẽ đăng ký với kênh sự kiện. Khi một tiến trình thành viên có yêu cầu dịch vụ, nó gửi đến kênh sự kiện, kênh sự kiện sẽ tiếp nhận và tìm kiếm tiến trình có thể đáp ứng yêu cầu và kết nối các thành viên đó để trao đổi với nhau.
  + **Ví dụ ứng dụng:** Rất phù hợp để xây dựng các hệ thống có khả năng phân tải và dự phòng nóng, nên sử dụng CORBA hoặc JAVA RMI làm nền tảng cho các ứng dụng này.
* **Mô hình dữ liệu tập trung (Centralized Data Model):**
  + **Mô tả:** Ba mô hình đã đề cập trên đòi hỏi các tiến trình cung cấp dịch vụ luôn phải ở trạng thái hoạt động, hơn nữa nếu lượng dữ liệu lớn có thể dẫn tới hiện tượng quá thời gian.

**-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:**

* **Vai trò:** Phần mềm trung gian đóng vai trò quan trọng trong việc đơn giản hóa cách phát triển hệ thống thông tin phân tán. Nó nhận các yêu cầu truy vấn và quyết định gửi yêu cầu đến đâu (ví dụ: khi dữ liệu được nhân bản).
* **Ba tính năng chính mà nó cung cấp:**
  + **Đơn giản hóa phát triển hệ thống:** Bằng cách trừu tượng hóa sự phức tạp của việc phân tán dữ liệu và xử lý.
  + **Quản lý truy nhập dữ liệu:** Đặc biệt là khi dữ liệu được nhân bản, middleware sẽ quyết định cách thức truy vấn và cập nhật để đảm bảo tính nhất quán.
  + **Che giấu tính không đồng nhất:** Middleware giúp che giấu sự khác biệt về phần cứng, hệ điều hành và giao thức mạng, tạo ra một môi trường đồng nhất hơn cho các ứng dụng.

**Câu 6:**

- Kiến trúc hướng dịch vụ chia các dịch vụ thành dịch vụ cơ bản, dịch vụ tích hợp và dịch vụ qui trình, ba loại này liên quan mật thiết đến quá trình cung cấp dịch vụ. Dịch vụ cơ bản cung cấp các tính năng kinh doanh cơ bản nhất, chúng chưa được phân cho các dịch vụ khác, thời gian chạy tương đối ngắn và thuộc loại không trạng thái, do đó các dịch vụ này rất phù hợp với phương thức gọi đồng bộ.

Thực tế, các dịch vụ cơ bản thường được cài đặt để truy nhập dữ liệu và một số nghiệp vụ cơ bản như tạo một người dùng mới hoặc thay đổi mật khẩu… Các dịch vụ tích hợp được cấu thành từ một số dịch vụ cơ bản, nhìn chung thời gian thực hiện của các dịch vụ này cũng tương đối ngắn và thuộc loại không trạng thái, sự tích hợp ở đây có thể thuộc về một hoặc nhiều nền tảng. Dịch vụ qui trình khác với hai loại trên, nó phản ánh một qui trình kinh doanh, do đó thời gian thực hiện khá dài và thuộc loại có trạng thái.

-Trục dịch vụ doanh nghiệp là hạ tầng kiến trúc cho phép sử dụng các dịch vụ của hệ thống sản xuất, thường triển khai các ứng dụng, các nền tảng và các quy trình nghiệp vụ. Các dịch vụ này được liên kết và trao đổi thông tin với nhau nhưng không sử dụng một loại định dạng dữ liệu chung và cũng không có một chuẩn giao tiếp chung. Nếu cần giao tiếp với các hệ thống bên ngoài, vấn đề tích hợp sẽ mở rộng ra khỏi phạm vi của doanh nghiệp, nó bao chùm lên các hệ thống và quy trình nghiệp vụ của các doanh nghiệp khác nhau.

Những năm gần đây, một số giải pháp như tích hợp ứng dụng doanh nghiệp, doanh nghiệp với doanh nghiệp, kiến trúc hướng dịch vụ và dịch vụ Web đã tập trung giải quyết những vấn đề liên quan tới tích hợp hệ thống thông tin của các doanh nghiệp. Những giải pháp trên tập trung vào một vài vấn đề về tích hợp, nhưng chúng thường là sản phẩm của một công ty nào đó, giá thành đắt và tốn thời gian triển khai.

-Trục dịch vụ doanh nghiệp theo tiêu chuẩn sẽ giải quyết các vấn đề liên quan đến việc tích hợp mà không cần phải loại bỏ những giải pháp đang sử dụng. Mục đích của trục dịch vụ doanh nghiệp là làm cho việc tích hợp các ứng dụng và quy trình trở nên thuận tiện hơn bằng cách cung cấp một quy trình phân tán, điều hướng thông minh, bảo mật và có thể tự động chuyển đổi dữ liệu. Trong trục dịch vụ doanh nghiệp, những dịch vụ trên là những dịch vụ nền tảng do đó các ứng dụng không cần phải thi hành riêng biệt những yêu cầu trên theo một cách thức riêng biệt của chúng.

Trục dịch vụ doanh nghiệp giải quyết những điểm yếu của những giải pháp có sẵn bằng cách tạo ra một nền tảng chuẩn cho việc tích hợp. Giải pháp điểm–điểm yêu cầu cứ N thành phần tham gia hệ thống thì phải có N-1 giao diện để có thể giao tiếp được với các thành phần còn lại được thay thế bằng giải pháp trục, mỗi thành phần chỉ cần một giao diện để giao tiếp với trục và như vậy sẽ giao tiếp với các thành phần còn lại. Trục dịch vụ doanh nghiệp đảm bảo giao tiếp phân tán, chuyển hướng, xử lý nghiệp vụ, ổn định và bảo mật, đồng thời cũng cung cấp các dịch vụ có khả năng cắm và chạy.

Xây dựng hệ thống dựa trên SOA thoạt nhìn khá phức tạp, nó tùy thuộc vào góc nhìn đối với hệ thống, có thể đó là góc độ kinh doanh hay kỹ thuật. Mô hình logic phân chia hệ thống thành các miền, mỗi miền đảm nhiệm vai trò và trách nhiệm riêng, khái niệm miền ở đây phản ánh một thực thể nào đó, ví dụ đó là công ty, phòng/ban… Đứng trên góc độ kỹ thuật có thể thấy trục dịch vụ doanh nghiệp đóng vai trò trung tâm, các vùng chỉ cung cấp các dịch vụ cơ bản và dịch vụ tích hợp, các dịch vụ qui trình được tách biệt riêng rẽ.

So với kiến trúc dựa trên thành phần, điểm khác biệt chính của SOA là cung cấp khả năng giao tiếp giữa các dịch vụ sử dụng thông điệp dựa trên các giao thức phổ biến như HTTP, FTP, SMTP, .... và vì vậy kiến trúc SOA mới có khả năng độc lập với nền tảng. Các dịch vụ hoạt động trên nền tảng khác nhau vẫn có thể giao tiếp với nhau nhờ vào các giao diện đã được chuẩn hóa để cộng tác xử lý một tác vụ nào đó.

Phương thức trao đổi thông điệp đã được tất cả các nền tảng và ngôn ngữ lập trình hỗ trợ, do đó các dịch vụ trên các nền tảng nào sẽ hoạt động với cấu trúc dữ liệu đặc thù

của nền tảng đó. Trao đổi thông điệp có thể thực hiện theo cơ chế không đồng bộ, bên gửi và và bên nhận không cần phải chờ nhau, điều này giúp cho mỗi bên tiếp tục xử lý công việc sau khi gởi thông điệp mà không cần dừng thực thi để chờ thông điệp trả lời.