Được phân phối Hệ thống cấu trúc

Bài tập thực hành

16.1Hầu hết các mạng WAN chỉ sử dụng cấu trúc liên kết được kết nối một phần. Tại sao cái này rất?

Trả lời: Trị giá. Một mạng được kết nối đầy đủ yêu cầu liên kết giữa mọi nút trong mạng. Đối với mạng WAN, điều này có thể quá tốn kém vì các liên kết giao tiếp giữa các máy chủ ở xa về mặt vật lý có thể tốn kém.

16,2 Trong những trường hợp nào thì mạng truyền mã thông báo hiệu quả hơn mạng Ethernet?

Trả lời:Vòng mã thông báo rất hiệu quả trong điều kiện tải duy trì cao, vì không có xung đột nào có thể xảy ra và mỗi vị trí có thể được sử dụng để mang một thông điệp, cung cấp thông lượng cao. Vòng mã thông báo kém hiệu quả hơn khi tải nhẹ (xử lý mã thông báo mất nhiều thời gian hơn so với truy cập xe buýt, vì vậy bất kỳ gói nào có thể mất nhiều thời gian hơn để đến đích) hoặc rời rạc.

16.3Tại sao việc các cổng truyền các gói tin quảng bá giữa các mạng lại là một ý tưởng tồi? Lợi ích của việc làm như vậy là gì?

Trả lời:Tất cả các chương trình phát sóng sẽ được truyền tới tất cả các mạng, gây ra*nhiều*lưu lượng mạng. Nếu lưu lượng truyền phát bị giới hạn ở dữ liệu quan trọng (và rất ít trong số đó), thì việc truyền quảng bá sẽ giúp các cổng không phải chạy phần mềm đặc biệt để theo dõi dữ liệu này (chẳng hạn như thông tin định tuyến mạng) và phát lại dữ liệu đó.

16.4 Thảo luận về những ưu điểm và nhược điểm của các bản dịch tên bộ nhớ đệm cho các máy tính đặt trong các miền từ xa.

Trả lời:Có một lợi thế về hiệu suất đối với bản dịch tên bộ đệm cho các máy tính nằm trong miền từ xa: độ phân giải lặp lại của cùng một tên từ các máy tính khác nhau nằm trong miền cục bộ có thể được thực hiện cục bộ mà không yêu cầu thao tác tra cứu tên từ xa. Điểm bất lợi là có thể có sự mâu thuẫn trong

bản dịch tên khi cập nhật được thực hiện trong ánh xạ tên tới địa chỉ IP. Những vấn đề về tính nhất quán này có thể được giải quyết bằng cách làm mất hiệu lực bản dịch, điều này sẽ yêu cầu trạng thái được quản lý liên quan đến máy tính nào đang lưu vào bộ nhớ đệm một bản dịch nhất định và cũng sẽ yêu cầu một số thông báo vô hiệu hoặc bằng cách sử dụng hợp đồng thuê theo đó thực thể bộ nhớ đệm làm mất hiệu lực bản dịch sau một khoảng thời gian nhất định của thời gian. Cách tiếp cận thứ hai yêu cầu ít trạng thái hơn và không có thông báo vô hiệu nhưng có thể bị mâu thuẫn tạm thời.

16,5 Ưu điểm và nhược điểm của việc sử dụng chuyển mạch kênh là gì? Đối với những loại ứng dụng nào, chuyển mạch kênh là một chiến lược khả thi?
Trả lời: Chuyển mạch kênh đảm bảo rằng các tài nguyên mạng cần thiết cho quá trình truyền được dự trữ trước khi quá trình truyền diễn ra. Điều này đảm bảo rằng các gói tin sẽ không bị rơi và việc phân phối của chúng sẽ đáp ứng các yêu cầu về chất lượng dịch vụ. Nhược điểm của chuyển mạch kênh là nó yêu cầu một bản tin khứ hồi để thiết lập các đặt chỗ và nó cũng có thể cung cấp quá mức tài nguyên, do đó dẫn đến việc sử dụng tài nguyên dưới mức tối ưu. Chuyển mạch kênh là một chiến lược khả thi cho các ứng dụng có nhu cầu liên tục về tài nguyên mạng và sẽ yêu cầu tài nguyên trong thời gian dài, do đó phân bổ chi phí chung ban đầu.

16,6 Hai vấn đề ghê gớm mà các nhà thiết kế phải giải quyết để triển khai một hệ thống mạng trong suốt là gì?
Trả lời: Một trong những vấn đề như vậy là làm cho tất cả các bộ xử lý và thiết bị lưu trữ có vẻ trong suốt trên toàn mạng. Nói cách khác, hệ thống phân tán nên xuất hiện như một hệ thống tập trung cho người dùng. Hệ thống tệp Andrew và NFS cung cấp tính năng này: hệ thống tệp phân tán xuất hiện với người dùng dưới dạng một hệ thống tệp duy nhất nhưng trên thực tế, nó có thể được phân phối trên một mạng.
Một vấn đề khác liên quan đến tính di động của người dùng. Chúng tôi muốn cho phép người dùng kết nối với "hệ thống" chứ không phải với một máy cụ thể (mặc dù trên thực tế, họ có thể đang đăng nhập vào một máy cụ thể ở đâu đó trong hệ thống phân tán).

16,7 Việc di chuyển quy trình trong một mạng không đồng nhất thường là không thể, do sự khác biệt về kiến trúc và hệ điều hành. Mô tả một phương pháp để di chuyển quy trình trên các kiến trúc khác nhau đang chạy:

Một. Hệ điều hành giống nhau

b. Hệ điều hành khác nhau

Trả lời: Đối với cùng một hệ điều hành, quá trình di chuyển tương đối đơn giản, vì trạng thái của quá trình cần di chuyển từ bộ xử lý này sang bộ xử lý khác. Điều này liên quan đến việc di chuyển không gian địa chỉ, trạng thái của các thanh ghi CPU và các tệp đang mở từ hệ thống nguồn đến đích. Tuy nhiên, điều quan trọng là các bản sao giống hệt nhau của hệ điều hành đang chạy trên các hệ thống khác nhau để đảm bảo tính tương thích. Nếu hệ điều hành giống nhau, nhưng có lẽ các phiên bản khác nhau đang chạy trên các hệ thống riêng biệt, thì quá trình di chuyển phải

đảm bảo tuân theo các nguyên tắc lập trình nhất quán giữa các phiên bản khác nhau của hệ điều hành.
Các ứng dụng Java cung cấp một ví dụ hay về quá trình di chuyển giữa các hệ điều hành khác nhau. Để che giấu sự khác biệt trong hệ thống cơ bản, quá trình đã di chuyển (tức là một ứng dụng Java) chạy trên một máy ảo chứ không phải một hệ điều hành cụ thể. Tất cả những gì cần thiết là để máy ảo đang chạy trên hệ thống mà tiến trình di chuyển đến.

16.8 Để xây dựng một hệ thống phân tán mạnh mẽ, bạn phải biết những loại lỗi nào có thể xảy ra.

Một. Liệt kê ba loại lỗi có thể xảy ra trong hệ thống phân tán.

b. Chỉ định mục nào trong danh sách của bạn cũng có thể áp dụng cho hệ thống tập trung.

Trả lời: Ba lỗi phổ biến trong hệ thống phân tán bao gồm: (1) lỗi liên kết mạng, (2) lỗi máy chủ, (3) lỗi phương tiện lưu trữ. Cả (2) và (3) đều là lỗi cũng có thể xảy ra trong hệ thống tập trung, trong khi lỗi liên kết mạng chỉ có thể xảy ra trong hệ thống được phân phối theo mạng.

16,9 Có phải luôn luôn quan trọng để biết rằng tin nhắn bạn đã gửi đã đến đích một cách an toàn? Nếu câu trả lời của bạn là *Vâng,* giải thích vì sao. Nếu câu trả lời của bạn là *không,* đưa ra các ví dụ thích hợp.
Trả lời: Không. Nhiều chương trình thu thập trạng thái hoạt động dựa trên giả định rằng hệ thống đích có thể không nhận được các gói. Các chương trình này nói chung *phát tin* một gói tin và giả định rằng ít nhất một số hệ thống khác trên mạng của họ sẽ nhận được thông tin. Ví dụ: một daemon trên mỗi hệ thống có thể phát đi số lượng người dùng và mức trung bình tải của hệ thống. Thông tin này có thể được sử dụng để lựa chọn mục tiêu di chuyển quy trình. Một ví dụ khác là một chương trình xác định xem một trang web từ xa có đang chạy và có thể truy cập được qua mạng hay không. Nếu nó gửi một truy vấn và không nhận được phản hồi, nó biết rằng hệ thống hiện không thể kết nối được.

16,10 Trình bày một thuật toán để tái tạo lại một vòng logic sau khi một quá trình trong vòng bị lỗi.
Trả lời: Các hệ thống phân tán thông thường sử dụng một quy trình điều phối viên thực hiện các chức năng cần thiết của các quy trình khác trong hệ thống. Điều này sẽ bao gồm việc thực thi loại trừ lẫn nhau và — trong trường hợp này là một chiếc nhẫn - thay thế một mã thông báo bị mất.
Có thể sử dụng một lược đồ tương tự như thuật toán vòng được trình bày trong Phần 18.6.2. Thuật toán như sau:
Các **thuật toán vòng** giả định rằng các liên kết là đơn hướng và các quá trình sẽ gửi thông điệp của chúng đến người hàng xóm ở bên phải của chúng. Cấu trúc dữ liệu chính được sử dụng bởi thuật toán là **danh sách hoạt động**, một danh sách chứa các số ưu tiên của tất cả các tiến trình đang hoạt động trong hệ thống khi thuật toán kết thúc; mỗi quy trình duy trì danh sách hoạt động của riêng nó.

Một. Nếu quá trình *Ptôi* phát hiện lỗi điều phối viên, nó tạo ra một danh sách hoạt động mới mà ban đầu trống. Sau đó nó sẽ gửi một tin nhắn *trúng tuyển*(*tôi*) cho hàng xóm bên phải của nó và thêm số *tôi* vào danh sách hoạt động của nó.

b. Nếu $P_{tôi}$nhận được một tin nhắn *trúng tuyển*(*j*) từ quy trình bên trái, nó phải phản hồi theo một trong ba cách:

1. Nếu đây là lần đầu tiên*trúng tuyển*tin nhắn nó đã thấy hoặc đã gửi, $P_{tôi}$ tạo một danh sách hoạt động mới với các số*tôi*và*j*.Sau đó nó sẽ gửi tin nhắn*trúng tuyển*(*tôi*), tiếp theo là tin nhắn*trúng tuyển*(*j*).
2. Nếu*tôi*- =*j*, nghĩa là, tin nhắn nhận được không chứa $P_{tôi}$'số s, sau đó $P_{tôi}$ thêm vào*j*vào danh sách hoạt động của nó và chuyển tiếp tin nhắn đến người hàng xóm bên phải của nó.
3. Nếu*tôi*=*j*, đó là, $P_{tôi}$nhận được tin nhắn*trúng tuyển*(*tôi*), sau đó là danh sách hoạt động cho $P_{tôi}$bây giờ chứa số của tất cả các quy trình đang hoạt động trong hệ thống. Quy trình $P_{tôi}$bây giờ có thể xác định số lượng lớn nhất trong danh sách hoạt động để xác định quy trình điều phối viên mới.

16.11Hãy xem xét một hệ thống phân tán với hai trang web, A và B. Hãy xem xét liệu trang web A có thể phân biệt giữa các trang sau:

Một. B đi xuống.

b. Liên kết giữa A và B đi xuống.

C. B cực kỳ quá tải và thời gian phản hồi của nó lâu hơn bình thường 100 lần.

Câu trả lời của bạn có ý nghĩa gì đối với việc khôi phục trong hệ thống phân tán?

Trả lời:Một kỹ thuật sẽ dành cho B để định kỳ gửi một*Tôi đang lên* thông báo cho A cho biết nó vẫn còn sống. Nếu A không nhận được một*Tôi rất vui*, nó có thể giả sử B — hoặc liên kết mạng — bị ngắt. Lưu ý rằng một*Tôi đang lên*thông báo không cho phép A phân biệt giữa từng loại hư hỏng. Một kỹ thuật cho phép A tốt hơn để xác định xem mạng có bị trục trặc hay không là gửi một*Bạn đang lên*nhắn tin cho B bằng một con đường thay thế. Nếu nó nhận được phản hồi, nó có thể xác định rằng thực sự liên kết mạng bị ngắt và B đã lên.

Nếu chúng ta giả định rằng A biết B đã lên và có thể truy cập được (thông qua*Tôi rất vui*cơ chế) và A có một số giá trị*n*cho biết thời gian phản hồi bình thường, A có thể theo dõi thời gian phản hồi từ B và so sánh các giá trị với*n*, cho phép A xác định xem B có quá tải hay không. Ý nghĩa của cả hai kỹ thuật này là A có thể chọn một máy chủ khác — chẳng hạn như C — trong hệ thống nếu B gặp sự cố, không thể truy cập hoặc quá tải.