Câu 1: Chỉ ra định nghĩa đúng nhất của Hệ Phân Tán

1. **Tập hợp các máy tính độc lập có kết nối với nhau, cung ứng dịch vụ cho người sử dụng như 1 máy tính đơn nhất**
2. Tập hợp các máy tính ko có kết nối với nhau, cung ứng dịch vụ cho NSD như 1 máy tính đơn duy nhất
3. Tập hợp các máy tính đồng nhất có kết nối với nhau, cung cấp cung cấp dịch vụ cho NSD như 1 máy tính đơn duy nhất
4. Tập hợp các máy tính độc lập có kết nối với nhau, cung cấp ứng dụng cho người sử dụng như 1 mạng cục bộ nhiều máy tính

**Key = A**

Câu 2: Một hệ thống áp dụng thuật toán tập trung cho các giải thuật đồng bộ hoá loại trừ lẫn nhau, biết rằng hệ thống có n tiến trình, trong đó có 1 tiến trình làm Coordinator. Hỏi số thông điệp cần thiết phải sử dụng để cho k tiến trình vào sử dụng tài nguyên là bao nhiêu? (gs cả quá trình ko có lỗi, và không có 2 yêu cầu nào đến Coordinator vào cùng 1 thời điểm)

**Key = 3\*k**

Câu 3: Người ta gọi hệ thống phân tán hợp tác (Collaborative Distributed Systems) là hệ thống có kiến trúc hỗn hợp là vì

**Key = Kết hợp kiến trúc tập trung và không tập trung**

Câu 4: Liên quan đến vấn đề phân phối nội dung, nên chọn giải pháp cập nhật dữ liệu nào để phù hợp với dịch vụ DNS và WWW

**Key = Truyền dữ liệu cập nhật**

Câu 5: Đâu là sự khác biệt giữa cơ chế đồng bộ hoá vật lý và đồng bộ hoá logic

**Key = Đồng bộ hoá vật lý hướng tới hiệu chỉnh đồng hồ các máy về giống nhau, còn đồng bộ hoá logic chỉ quan tâm tới thứ tự sự kiện**

Câu 6: Hệ thống CAN là một hệ thống

**Key = Có kiến trúc không tập trung, có cấu trúc**

Câu 7: Một hệ thống có 16 tiến trình P0 – P15, áp dụng giải thuật bầu chọn Bully để chọn ra 1 tiến trình làm Coordinator. GS có 3 tiến trình bị lỗi dừng hđ là P13, P14, P15. GS P6 bắt đầu phát động bầu chọn. Hỏi hệ thống mất bao nhiêu thông điệp để chọn ra Coordinator = P12 (tính cả thông điệp quảng bá của P0 – P11)

**Key = 75**

Câu 8: Đâu là nhược điểm của giải pháp định danh dựa vào Home Agent

**Key = Mỗi khi muốn liên lạc với mọi thực thể, client phải liên lạc trước tiên với home agent dù ở rất xa**

Câu 9: Trong cơ chế sao lưu dựa trên túc số (quorum), điều kiện Nw>N/2 để làm gì

**Key = Tránh mâu thuẫn ghi-ghi**

Câu 10: RabbitMQ là công cụ tiện lợi và hiệu quả trong xây dựng

**Key = Cơ chế trao đổi thông tin hướng thông điệp bền vững**

Câu 11: Giao thức ARP dùng cơ chế định danh nào

**Key = Quảng bá**

PP định danh trong hệ thông Chord sử dụng DST. GS một hệ thống Chord với các bảng như hình dưới đây. Trả lời 3 câu hỏi 12-14

Câu 12: Các giá trị của bảng finger table của nút 5 là

**Key = 8, 8, 11, 2**

Câu 13: Các giá trị của bảng finger table của nút số 8 là

**Key = 11, 11, 2, 2**

Câu 14: Thứ tự các nút mà hệ thống phải duyệt qua để tìm khoá số 9 từ nút số 2 là (biết rằng chiều tìm kiếm là chiều kim đồng hồ)

**Key = 8 -> 11**

Câu 15: Socket UDP và socket TCP khác nhau ntn

**Key = Socket UDP sử dụng cơ chế multiplexing, như vậy server sẽ ko cần tạo socket với mỗi yêu cầu kết nối. Còn socket TCP thì ngược lại, sẽ tạo 1 socket mới cho từng yêu cầu kết nối**

Câu 16: Trong các kiểu kiến trúc server đa luồng, xét kiến trúc luồng cho mỗi yêu cầu (thread-per-request). GS ứng với mỗi yêu cầu thì thời gian tiêu tốn cho luồng thực thi tương ứng đều như nhau và là 3ms để khởi tạo luồng, 5ms để thực thi công việc, 2ms để huỷ luồng. Server đó nhận được n yêu cầu từ các client gửi tới vào 2 thời điểm. Thời điểm đầu nhận đc cùng lúc n-k yêu cầu, thời điểm sau nhận đc cùng lúc k yêu cầu. Hỏi server mất tổng bao nhiêu thời gian để xử lý n yêu cầu trên

**Key = 20**

Câu 17: Bài toán yêu cầu bạn so sánh thời gian đọc một tệp (file) của một máy chủ tệp tin (file server) đơn luồng và một máy chủ đa luồng. Phải mất tổng cộng 24ms để nhận 1 yêu cầu (resquest) và thực hiện quá trình xử lý, giả định rằng các dữ liệu cần thiết nằm ở bộ nhớ đệm trong bộ nhớ chính. Nếu cần thiết phải thực hiện một thao tác truy cập ổ đĩa thì cần thêm 90ms, biết rằng việc phải thực hiện thao tác này có xác suất là 1/4. Hỏi máy chủ có thể nhận đc bao nhiêu yêu cầu/giây trong 2 trường hợp: máy chủ là đơn luồng và máy chủ là đa luồng (ngoài luồng nhận và xử lý request, sẽ có thêm 1 luồng để truy cập ổ đĩa nếu cần thiết)?

**Key = Đơn luồng 21.5 yêu cầu/s; Đa luồng 41.67 yêu cầu/s**

Câu 18: Vai trò của Middleware trong các HPT

**Key = Cung cấp tính trong suốt cho các ứng dụng phân tán ở tầng trên**

Câu 19: Vấn đề gặp phải trong cơ chế RPC khi truyền tham số là tham biến (tham số giá trị)

**Key = Máy client và máy server có qui ước biểu diễn dữ liệu khác nhau**

Với việc sử dụng Vector Clock để thực thi trao đổi thông tin có tính nhân quả, GS có các tiến trình trao đổi thông điệp với nhau như hình dưới đây, trả lời câu 20 và 21

Câu 20: Thông điệp nào sẽ bị làm trễ lại (ở tầng Middleware) khi đẩy thông tin lên tầng ứng dụng vì không thoả mãn đk trao đổi thông tin có tính nhân quả

**Key = Thông điệp m1 từ P0 -> P2**

Câu 21: GS thời điểm ban đầu các giá trị vector clock đều là (0, 0, 0, 0), vậy thì các giá trị vector clock ở các thời điểm T1 – T4 là

**Key = T1=(0, 1, 0, 0); T2=(1, 1, 0, 0); T3=(0, 0, 0, 0); T4=(1, 0, 0, 0)**

Câu 22: Xem xét cơ chế định danh để xác định vị trí của thực thể trong không gian tên phẳng, người ta chọn giải pháp tiếp cận phân cấp. Khi một thực thể chuyển động từ vị trí A sang B, vấn đề gặp phải đối với hệ thống là ở các vị trí trung gian thì thời gian dừng lại của thực thể đó là rất ngắn, nếu thực hiện đk thì vị trí trung gian đó rất lãng phí tài nguyên, hơn nữa thời gian để đk vị trí trung gian đó lớn hơn thời gian tạm dừng của thực thể. Vậy giải pháp là

**Key = Ở các vị trí trung gian, kết hợp áp dụng pp chuyển tiếp con trỏ**

Một hệ thống hướng tới đảm bảo thống nhất theo mô hình thống nhất liên tục, nhà quản trị sử dụng Conit để theo dõi 3 biến x, y, z. GS có 2 máy chủ bản sao (replica) A và B với các thao tác cập nhật diễn ra như hình dưới đây. Các thao tác in màu xám là đã đc committed. Có 1 thao tác cập nhật ở B đã đc phổ biến sang A và ngược lại có một thao tác đã đc phổ biến từ A sang B. Trả lời câu 23-25

Câu 23: Vector Clock đại diện cho chênh lệch liên quan đến thời gian ở mỗi replica là bn

Key =

Câu 24: Độ lệch về thứ tự là bn

**Key = A là 3, B là 3**

Câu 25: Cặp giá trị đại diện cho độ lệch về giá trị là

Key =

Câu 26: DNS ngày nay chủ yếu ko hỗ trợ phân giải tên miền đệ quy, đặc biệt là các máy chủ DNS cấp cao. Vậy có nên chuyển hết chúng sang chạy hỗ trợ pp phân giải tên miền đệ quy ko

**Key = Ko vì liên quan đến 2 lý do là hiệu năng và ATTT (phòng chống tấn công dựa vào điểm yếu của phân giải đệ quy)**

Câu 27: Có một mô hình với 5 tiến trình P1 – P5 thực hiện các thao tác đọc ghi lên các bản ghi cục bộ (tất cả bản ghi tạo thành kho dữ liệu phân tán) như hình vẽ. Phát biểu nào sau đây đúng

**Key = MH thoả mãn thống nhất nhân quả và ko thoả thống nhất tuần tự; Key2 = ko thoả cả 2**

Câu 28: Đâu là nhược điểm nếu như ta đưa quá nhiều biến vào Conit để giám sát độ lệch giữa các máy chủ bản sao?

**Key = Làm cho các replica nhanh chóng rơi vào trạng thái ko thống nhất**

Câu 29: Hệ thống áp dụng TT Token Ring, giải thuật đồng bộ loại trừ lẫn nhau. Hệ thống có n tiến trình và GS thời gian để chuyển thẻ bài giữa 2 nút liền kề là x ms, thời gian để 1 nút vào sử dụng tài nguyên là y ms. Vậy một nút chuyển tiếp token đi phải chờ lâu nhất bao nhiêu lâu mới nhận lại token (GS hệ thống ko lỗi)

**Key = (n-1)\*y + n\*x**

Câu 30: Một hệ thống triển khai đảm bảo thống nhất dữ liệu hướng người dùng. Xét một client dịch chuyển giữa các máy chủ bản sao (L1-L4) như hình dưới đây, hãy cho biết mô hình này ko thoả mãn mô hình thống nhất hướng dữ liệu nào

**Key = Đọc đơn điệu**

Câu 31: Xét các giao thức dựa vào bản sao primary trong nhóm các giao thức đảm bảo thống nhất, giao thức ghi từ xa khác giao thức ghi cục bộ ntn

**Key = Giao thức ghi từ xa thì các thao tác ghi sẽ đc gửi tới bản sao primary của thành phần dữ liệu tương ứng, còn giao thức ghi cục bộ thì thực hiện ghi luôn trên bản sao cục bộ của tiến trình ghi**

Câu 32: Để triển khai một cluster sử dụng Kubernetes, ngoài việc tự cấu hình cho từng máy thì có một công cụ tên là minikube giúp chúng ta triển khai nhanh chóng hơn. Minikube có tác dụng gì

**Key = Tạo ra 1 cluster ảo trong đó có một woker node và 1 master node**

Câu 33: Nodejs là 1 flatform hiệu quả trong xây dựng server và các ứng dụng mạng. Hãy chọn phát biểu sai

**Key = Nodejs sử dụng mô hình đa luồng**

Câu 34: Chord là hệ thống

**Key = Có kiến trúc ko tập trung, có cấu trúc**

Câu 35: Khi cấu hình các DNS server cho 1 zone, ngoài primary DNS server thì người ta cần xây dựng cấu hình ít nhất cho 1 secondary DNS server vì

**Key = Secondary server sẽ thay thế cho primary server trong trường hợp bị hỏng, ngừng hoạt động, đồng thời giảm tải cho primary server**

Câu 36: Xét giải thuật phân tán (distributed algorithm) trong nhóm các giải thuật bù trừ lẫn nhau (mutual exclusion), nếu có một tiến trình A muốn vào sử dụng tài nguyên mà cũng nhận đc 1 thông điệp yêu cầu vào sử dụng tài nguyên của 1 tiến trình B khác thì tiến trình A sẽ làm gì

**Key = Kiểm tra timestamp trên yêu cầu của A, B xem timestamp nào nhỏ hơn thì thắng, tiến trình đó đc vào sử dụng tài nguyên**

Câu 37: Để đảm bảo đồng bộ hoá trong lập trình song song (ngôn ngữ C), người ta sử dụng kỹ thuật mutex lock. Kỹ thuật này chia làm 2 pp là Coarse locking và Fine locking. Điểm khác nhau của 2 pp trên

**Key = Fine locking tạo ra đoạn găng (critical section) riêng cho mỗi tài nguyên, Coarse locking tạo ra một đoạn duy nhất cho tất cả tài nguyên**

Câu 38: GS hệ thống có n máy, mất bn thông điệp cho một lần thực hiện đồng bộ hoá sử dụng giải thuật Berkeley

**Key = 3n**

Câu 39: Một hệ thống áp dụng thuật toán phân tán (Distributed algorithm) cho các giải thuật đồng hộ hoá loại trừ lần nhau (mutual exclusion), biết rằng hệ thống có n tiến trình. Hỏi số thông điệp cần thiết phải sử dụng để cho k tiến trình vào sử dụng tài nguyên là bao nhiêu? (GS quá trình ko xảy ra lỗi và các yêu cầu đc gắn timestamp khác nhau)

**Key = 2\*k\*(n-1)**

Câu 40: Hệ thống quản lý tệp Linux vận hành dựa trên hệ thống định danh cho ko gian tên có cấu trúc. Trong 2 pp tạo liên kết soft-link và hard-link, đâu là pp làm tăng con đếm tham chiếu lên inode của file gốc

**Key = hard-link**