

# CHƯƠNG 7: Sao lưu và thống nhất dữ liệu

Họ tên sinh viên: Phan Thành Đạt

MSSV:20173001

Mã Lớp: 118636

Mã học phần: IT4611

## Câu 1:

Phải sao lưu dữ liệu vì:

- Tăng tính sẵn sàng: backup lẫn nhau nếu có bản sao bị hỏng, bị treo
- Cải thiện hiệu năng của hệ thống: có thể chọn bản sao gần
- Tăng tính chịu lỗi của hệ thống: phân tải cho các bản sao
- Tăng chất lượng dịch vụ và trải nghiệm người dùng

## Câu 2:

Câu 2				
P1	W(a)			
P2		W(b)		
P3	R(a)	W(c)		
P4			R(a)	R(b)
P5		R(b)	R(b)	R(c)

- a) Mô hình trên thỏa mãn thống nhất nhân quả vì quá trình W(a) ở P1 và W(c) P2 có tồn tại R(a) ở P2 nên giữa W(a) và W(c) là các sự kiện có tính nhân quả, vì thế cần được các tiến trình khác nhìn thấy theo cùng 1 thứ tự, tiến trình P4 và P5 đều thỏa mãn việc R(a) rồi mới R(c). Có thể quá trình ghi b vào P4 trước khi P4 đọc nên sẽ không thấy được R(a). Ngoài ra các sự kiện không có tính nhân quả thì không cần thứ tự.
- b) Mô hình toàn cục : W(a)R(a)W(b)R(b)W(c)R(c), do đó thỏa mãn mô hình thống nhất tuần tự.

## Câu 3:

Conit dùng để định nghĩa độ không thống nhất, từ đó để đánh giá độ thống nhất của dữ liệu. Cụ thể conit là tập các đơn vị dữ liệu của kho dữ liệu phân tán, mức độ thống nhất được xác định theo nhiều mức độ.

Nếu đặt kích thước conit quá lớn thì sẽ tăng số lượng cập nhật mỗi lần, dẫn đến sai lệch cao và chi phí cao, đặc biệt không hiệu quả đối với các dữ liệu độc lập, do mỗi conit sẽ có nhiều đơn vị dữ liệu nhưng bù lại sẽ có ít số conit.

Kích thước conit nhỏ sẽ làm giảm số lần cập nhật, giảm khả năng xuất hiện sai lệch tuy nhiên sẽ tạo ra rất nhiều conit và khó khăn cho vấn đề quản lý vì có quá nhiều conit.

#### Câu 4:

Thông nhất nhân quả có tính thống nhất yếu hơn thống nhất tuần tự vì đối với các thao tác ghi tương tranh hoặc song song thì chắc chắn sẽ có tính thống nhất nhân quả, tức là không cần yêu cầu các thao tác ghi tương tranh thống nhất thứ tự toàn cục từ đó sẽ không thỏa mãn thống nhất tuần tự.

#### Câu 5:

Vấn đề của mô hình Eventual Consistency:

- Chất lượng dịch vụ thấp với real-time
- Nếu tiến trình khác truy cập nhiều bản sao khác nhau thì vấn đề về thống nhất dữ liệu do các thao tác chưa kịp cập nhật đến bản sao truy cập trong khác các bản sao khác không được sử dụng bao giờ thì lại được cập nhật.

Định nghĩa mô hình thống nhất client: Chỉ cập nhật bản sao mà tiến trình cần sử dụng dữ liệu, các bản sao khác thì thôi. Đảm bảo các tiến trình dùng dữ liệu có 1 khung nhìn thống nhất về dữ liệu.

#### Câu 6:

- a) Đặt  $k - 1$  máy chủ sao cho tổng khoảng cách từ  $k - 1$  máy chủ đó đến mỗi chi nhánh là gần nhất, đặt máy chủ thứ  $k$  còn lại ở giữa các máy chủ  $k - 1$  sao cho khoảng cách từ máy chủ thứ  $k$  đến client là gần nhất.
- b) Nếu số lượng các yêu cầu tăng đối với đơn vị dữ liệu  $X$  vượt qua ngưỡng  $rep(x)$  thì server sẽ kích hoạt server bản sao cũng có khả năng phục vụ dữ liệu  $X$ . Nếu dưới ngưỡng  $del(x)$  thì server sẽ xóa dữ liệu  $X$  đó đi.
- c)

Sao lưu tích cực	Sao lưu tức số
Có 1 tiến trình chịu trách nhiệm phổ cập toàn bộ các thay đổi cho toàn bộ bản sao. Cần có 1 cơ chế cục bộ toàn cục, có thể dùng logic của Lamport, tuy nhiên không co giãn được	Thay vì cập nhật tất cả các bản sao thì chỉ cập nhật $N_w$ bản sao, và khi đọc thì đọc tối thiểu $N_r$ bản sao. Thỏa mãn 2 điều kiện $N_w + N_r > N$ $N_w > N/2$

#### Câu 7:

a)

Mô hình thống nhất hướng dữ liệu	Mô hình thống nhất hướng người dùng
Hướng đến đảm bảo dữ liệu thống nhất trên toàn bộ bản sao.	Không nhất thiết các bản sao phải có dữ liệu giống nhau mà đi đến bất kỳ đâu thì góc nhìn là duy nhất, thống nhất đối với từng tiến trình.

- b) Dịch vụ thư điện tử sẽ áp dụng mô hình thống nhất hướng dữ liệu vì đối với dịch vụ này thì dữ liệu của mỗi người chỉ phục vụ một người mà thôi nên khi tiến trình sử dụng bản sao nào thì sẽ cập nhật bản sao đó, không cần cập nhật cho toàn bộ bản sao.  
Dịch vụ www thì sẽ áp dụng mô hình thống nhất hướng người dùng vì đối với dịch vụ này thì dữ liệu sẽ có tính tương tranh, phục vụ nhiều tiến trình và cần phải cập nhật cho toàn bộ bản sao vì ứng dụng này có nhiều tiến trình dùng ở khắp nơi.
- c) Mô hình không tránh được xung đột đọc ghi vì  $N_w + N_r = 2700 < 3000$ . Mô hình tránh được xung đột ghi ghi vì  $N_w = 1600 > N/2 = 15000$