

## CHƯƠNG II

### CÁC VẤN ĐỀ VÀ GIẢI PHÁP CƠ BẢN TRONG CÁC HỆ PHÂN TÁN

#### NỘI DUNG

- Truyền thông
- Định danh
- Đồng bộ
- Tiến trình trong các hệ thống phân tán
- Quản trị giao dịch và điều khiển tương tranh
- Phục hồi và chịu lỗi
- Bảo mật
- Tính nhất quán và vấn đề nhân bản

#### TÍNH NHẤT QUÁN VÀ VẤN ĐỀ NHÂN BẢN

#### NỘI DUNG

- Giới thiệu nhân bản
- Mô hình lấy dữ liệu làm trung tâm
- Mô hình lấy máy khách làm trung tâm
- Quản lý bản sao
- Các giao thức đảm bảo tính toàn vẹn

#### NHÂN BẢN DỮ LIỆU

- Nhân bản dữ liệu để tăng độ tin cậy của hệ thống
- Nhân bản để nâng cao hiệu năng
  - Qui mô số lượng
  - Qui mô phạm vi địa lý
- Vấn đề nảy sinh:
  - Giảm tính nhất quán của dữ liệu
  - Chi phí tăng bằng thông để duy trì nhân bản
  - **Thao tác thực hiện nhân bản có thể làm giảm hiệu năng xử lý của hệ thống**

**Phải tốn nhiều công sức  
để xây dựng các mô hình đảm bảo  
tính nhất quán của dữ liệu.**

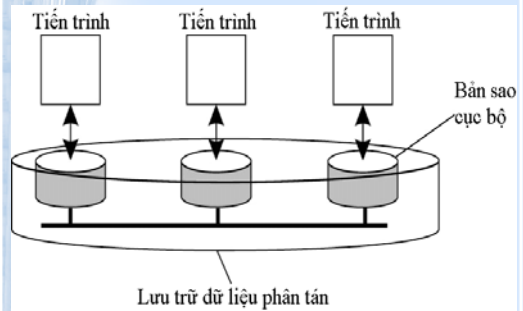
#### CÁC MÔ HÌNH NHÂN BẢN DỮ LIỆU

- Không có mô hình nhân bản dữ liệu tốt nhất. Phải thỏa hiệp giữa các tiêu chí nhân bản để đạt được mục tiêu yêu cầu.
- Mô hình lấy dữ liệu làm trung tâm
  - Mô hình nhất quán chặt
  - Mô hình nhất quán tuần tự
  - Mô hình nhất quán tuyến tính
  - Mô hình nhất quán nhân quả
  - Mô hình nhất quán FIFO
  - Mô hình nhất quán yếu
  - Mô hình nhất quán đi ra (release)
  - Mô hình nhất quán đi vào (entry)
- Mô hình nhất quán lấy máy khách làm trung tâm
  - Mô hình nhất quán cuối cùng
  - Mô hình nhất quán đọc đều
  - Mô hình nhất quán ghi đều
  - Mô hình nhất quán đọc kết quả ghi
  - Mô hình nhất quán ghi theo sau đọc

### MÔ HÌNH LẤY DỮ LIỆU LÀM TRUNG TÂM

- Tính nhất quán thường được hiểu trong ngữ cảnh các thao tác đọc/ghi bộ nhớ dùng chung (CSDL, tập tin...).
- Trong hệ thống phân tán, tính nhất quán được mở rộng cho trường hợp dữ liệu được lưu trữ tại các máy tính khác nhau.
- Tiến trình đọc dữ liệu từ bản sao (cục bộ) của dữ liệu hệ thống, thao tác ghi sẽ được lan truyền đến tất cả các bản sao dữ liệu trong thống.
- Khó khăn cơ bản trong việc nhân bản dữ liệu là thiếu cơ chế đồng hồ chung cho tất cả các máy tính trong hệ thống
- Hai vấn đề:
  - Tính nhất quán liên tục
  - Tính nhất quán thứ tự các thao tác

### MÔ HÌNH LẤY DỮ LIỆU LÀM TRUNG TÂM



### TÍNH NHẤT QUÁN LIÊN TỤC

- Nhân bản dữ liệu đưa ra vấn đề nhất quán, nhưng không có cách chung nào giải quyết hiệu quả vấn đề này. Chỉ có nói lòng tính nhất quán mới hy vọng đạt được giải pháp hiệu quả.
- Không có qui tắc chung nào cho việc nói lòng tính nhất quán, phụ thuộc lớn vào từng ứng dụng.
- Các cách tiếp cận nói lòng tính nhất quán:
  - Sự chênh lệch giá trị số giữa các bản sao: Giá trị có thể là của một trường dữ liệu nhưng cũng có thể là số lượng các thao tác thay đổi.
  - Sự chênh lệch trạng thái: Thời gian cập nhật cuối cùng
  - Sự chênh lệch thứ tự các thao tác cập nhật: Khá phức tạp, nhất là đối với các thao tác có ROLLBACK
- Sự chênh lệch trong ba cách tiếp cận trên hình thành phạm vi nhất quán liên tục

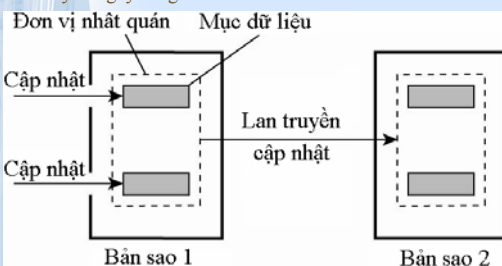
### VÍ DỤ NHẤT QUÁN LIÊN TỤC

Giả thiết ban đầu  $x=0, y=0$ . Độ lệch thứ tự thao tác của A là 3, độ lệch thứ tự thao tác của B là 2

Bản sao A		Bản sao B	
Đơn vị nhất quán		Đơn vị nhất quán	
$x = 6; y = 3$		$x = 2; y = 5$	
Thao tác	Kết quả	Thao tác	Kết quả
$\langle 5, B \rangle$	$x := x + 2$ [ $x = 2$ ]	$\langle 5, B \rangle$	$x := x + 2$ [ $x = 2$ ]
$\langle 8, A \rangle$	$y := y + 2$ [ $y = 2$ ]	$\langle 10, B \rangle$	$y := y + 5$ [ $y = 5$ ]
$\langle 12, A \rangle$	$y := y + 1$ [ $y = 3$ ]		
$\langle 14, A \rangle$	$x := y * 2$ [ $x = 6$ ]		

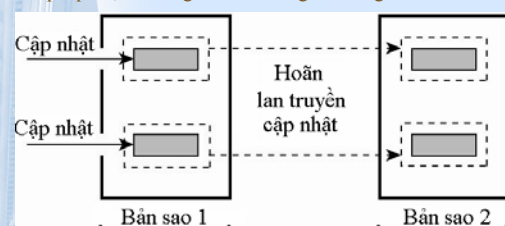
### CHỌN ĐƠN VỊ NHẤT QUÁN THÍCH HỢP

Thao tác cập nhật đã hoàn thành trên bản sao 1 phải được chuyển ngay sang bản sao 2



### CHỌN ĐƠN VỊ NHẤT QUÁN THÍCH HỢP

- Thao tác cập nhật đã hoàn thành trên bản sao 1 chưa cần chuyển ngay sang bản sao 2
- Chia nhỏ đơn vị nhất quán quá nhỏ không phải là giải pháp tốt, dễ làm giảm hiệu năng hệ thống



## TÍNH NHẤT QUÁN THEO THỨ TỰ CÁC THAO TÁC

- Hệ thống phân tán phải đương đầu với các công việc xử lý song song, tương tranh, chia sẻ tài nguyên...
- Nhiều mô hình đã được đưa ra, phần này sẽ đề cập tới mô hình đảm bảo tính nhất quán theo thứ tự các thao tác:
  - Mô hình nhất quán chặt
  - Mô hình nhất quán tuần tự
  - Mô hình nhất quán tuyến tính
  - Mô hình nhất quán nhân quả
  - Mô hình nhất quán FIFO
  - Mô hình nhất quán yếu
  - Mô hình nhất quán đi ra (release)
  - Mô hình nhất quán đi vào (entry)

## MÔ HÌNH NHẤT QUÁN CHẶT

- Định nghĩa: Bất kỳ thao tác đọc đều trả về một giá trị tương ứng với ghi **gần nhất** trên mục dữ liệu
- Các kí hiệu:
  - $W_i(x)a$ : thao tác ghi được thực hiện bởi tiến trình  $P(i)$  lên mục dữ liệu  $x$  với giá trị  $a$ .
  - $R_i(x)b$ : thao tác đọc được thực hiện bởi tiến trình  $P(i)$  lên mục dữ liệu  $x$  cho kết quả  $b$ .
  - Giả thiết  $x$  có giá trị ban đầu là null.

P1: $W(x)a$	P2: $W(x)a$
P2: $R(x)a$	P2: $R(x)NIL$ $R(x)a$

- Mô hình sử dụng khái niệm thời gian tuyệt đối (thời gian chung cho cả hệ thống để xác định đúng khái niệm **gần nhất**), điều này là khó khả thi với hệ phân tán.

## MÔ HÌNH NHẤT QUÁN TUẦN TỰ

- Kết quả thực hiện như nhau nếu các thao tác của các tiến trình được thực hiện theo thứ tự.
- Thao tác của mỗi tiến trình xuất hiện theo thứ tự đã được chương trình xác định.
- Khi các tiến trình chạy đồng thời trên các máy khác nhau thì cho phép sự đan xen của các thao tác nhưng tất cả các tiến trình đều phải nhận biết được sự đan xen của các thao tác.

Trên bản sao dữ liệu của các tiến trình P3, P4: Thao tác  $W(x)b$  thực hiện trước  $W(x)a$

P1: $W(x)a$	P2: $W(x)a$
P2: $W(x)b$	P2: $W(x)b$
P3: $R(x)b$ $R(x)a$	P3: $R(x)b$ $R(x)a$
P4: $R(x)b$ $R(x)a$	P4: $R(x)a$ $R(x)b$
Nhất quán tuần tự	Không thỏa mãn nhất quán tuần tự

Trên bản sao dữ liệu của tiến trình P4: Thao tác  $W(x)b$  thực hiện sau  $W(x)a$

P1: $W(x)a$	P2: $W(x)a$
P2: $W(x)b$	P2: $W(x)b$
P3: $R(x)b$ $R(x)a$	P3: $R(x)b$ $R(x)a$
P4: $R(x)b$ $R(x)a$	P4: $R(x)a$ $R(x)b$
Nhất quán tuần tự	Không thỏa mãn nhất quán tuần tự

## MÔ HÌNH NHẤT QUÁN TUẦN TỰ

- Giá trị ban đầu của các dữ liệu  $x, y, z$  đều bằng 0
- Giả sử ba tiến trình P1, P2, P3 thực hiện các thao tác:

Tiến trình P1	Tiến trình P2	Tiến trình P3
$x \leftarrow 1;$ $\text{print}(y, z);$	$y \leftarrow 1;$ $\text{print}(x, z);$	$z \leftarrow 1;$ $\text{print}(x, y);$

- Về lý thuyết sẽ có 720 tổ hợp xảy ra, nếu xét các thao tác gần trước thao tác in sẽ có 30 trường hợp.

## MÔ HÌNH NHẤT QUÁN TUẦN TỰ

- Kết quả: Giá trị được in ra
- Chữ ký: Kết quả in ra theo thứ tự tiến trình P1, P2, P3 yzxxzy

$x \leftarrow 1;$ $\text{print}(y, z);$ $y \leftarrow 1;$ $\text{print}(x, z);$ $z \leftarrow 1;$ $\text{print}(x, y);$	$x \leftarrow 1;$ $y \leftarrow 1;$ $\text{print}(x, z);$ $\text{print}(y, z);$ $z \leftarrow 1;$ $\text{print}(x, y);$	$y \leftarrow 1;$ $z \leftarrow 1;$ $\text{print}(x, y);$ $\text{print}(x, z);$ $x \leftarrow 1;$ $\text{print}(y, z);$	$y \leftarrow 1;$ $x \leftarrow 1;$ $z \leftarrow 1;$ $\text{print}(x, z);$ $\text{print}(x, y);$ $\text{print}(x, y);$
Kết quả: 001011 Chữ ký: 001011	Kết quả: 101011 Chữ ký: 101011	Kết quả: 010111 Chữ ký: 110101	Kết quả: 111111 Chữ ký: 111111

## MÔ HÌNH NHẤT QUÁN NHÂN QUẢ

- Mô hình này phân biệt các sự kiện có quan hệ nhân quả và các sự kiện không có quan hệ nhân quả.
- Nếu sự kiện  $b$  được gây ra hoặc bị tác động bởi một sự kiện  $a$  xảy ra sớm hơn thì mọi thực thể khác phải nhìn thấy  $a$  trước khi nhìn thấy  $b$ .
- Các thao tác ghi có quan hệ nhân quả tiềm năng phải được nhận biết bởi tất cả các tiến trình khác trong cùng một thứ tự. Các thao tác ghi đồng thời có thể nhận biết được theo thứ tự khác nhau trên các máy khác nhau.

P1: $W(x)a$	$W(x)c$		
P2: $R(x)a$	$W(x)b$		
P3: $R(x)a$		$R(x)c$	$R(x)b$
P4: $R(x)a$		$R(x)b$	$R(x)c$

### MÔ HÌNH NHẬT QUẢN NHÂN QUẢ

P1:	W(x)a		
P2:	R(x)a	W(x)b	
P3:		R(x)b	R(x)a
P4:		R(x)a	R(x)b

Ví phạm nhật quản nhân quả

P1:	W(x)a		
P2:		W(x)b	
P3:		R(x)b	R(x)a
P4:		R(x)a	R(x)b

Thứ tự chính xác của các sự kiện trong nhật quản nhân quả

### MÔ HÌNH NHẬT QUẢN FIFO

- Bộ qua giới hạn về trật tự của bất kì thao tác tương tranh nào.
- Các thao tác ghi bởi một tiến trình đơn phải được tất cả các tiến trình khác nhìn thấy theo cùng một trật tự mà chúng đề ra.
- Thao tác ghi bởi nhiều tiến trình khác nhau có thể được các tiến trình khác nhìn thấy theo những trật tự khác nhau.

P1:	W(x)a			
P2:	R(x)a	W(x)b	W(x)c	
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

### MÔ HÌNH NHẬT QUẢN YẾU

- Mô hình nhật quản yếu không tập trung vào các thao tác trên dữ liệu mà quan tâm đến trật tự các nhóm lệnh bằng việc sử dụng các biến đồng bộ hóa.
- Mô hình nhật quản yếu có ba đặc tính sau:
  - Việc truy cập đến một biến đồng bộ hóa được kết hợp với kho dữ liệu là một nhật quản tuần tự.
  - Không có thao tác nào lên các biến đồng bộ hóa được phép thực hiện cho đến khi tất cả các thao tác ghi trước đó được hoàn thành ở mọi nơi.
  - Không có thao tác đọc hay ghi dữ liệu lên các mục dữ liệu nào được phép thực hiện cho đến khi tất cả các thao tác trước đó lên các biến đồng bộ hóa được thực hiện.

### MÔ HÌNH NHẬT QUẢN ĐI RA (RELEASE)

- Sử dụng thêm hai lệnh: lệnh acquired để báo muốn vào vùng tới hạn (critical region) và lệnh release để báo giải phóng vùng tới hạn.
- Hai lệnh này cũng có hai cách thực thi khác nhau như: bằng một biến hoặc bằng một lệnh đặc biệt.
- Hai thao tác này chỉ thực hiện với các dữ liệu dùng chung chứ không áp dụng cho tất cả các dữ liệu.

### ĐIỀU KIỆN MÔ HÌNH NHẬT QUẢN ĐI RA

- Trước khi thực hiện một thao tác đọc hay ghi lên dữ liệu chia sẻ thì tất cả các thao tác acquire do tiến trình này thực hiện trước đó phải hoàn tất.
- Trước khi một thao tác release được phép thực hiện thì tất cả các thao tác đọc và ghi do tiến trình này thực hiện trước đó phải được hoàn tất.

P1:	Acq(L)	W(x)a	W(x)b	Rel(L)
P2:			Acq(L)	R(x)b
P3:				R(x)a

### MÔ HÌNH NHẬT QUẢN ĐI VÀO (Entry)

- Giống như mô hình nhật quản Release, mô hình nhật quản Entry sử dụng hai lệnh acquired và release khi muốn vào vùng tới hạn.
- Các lệnh này thao tác trên từng mục dữ liệu của vùng chia sẻ.
- Tiến trình nào muốn sử dụng mục dữ liệu thì phải đợi cho tất cả các tiến trình khác giải phóng mục dữ liệu đó.
- Để ghi lên một mục dữ liệu, máy khách phải có được biến đồng bộ hóa của mục đó trong chế độ dành riêng. Điều đó có nghĩa là không máy khách nào khác có thể sử dụng biến đó. Khi máy khách cập nhật xong mục dữ liệu, thì nó giải phóng biến đó.
- Khi máy khách muốn đọc một mục dữ liệu nào đó, nó phải có được biến đồng bộ hóa kết hợp ở chế độ không dành riêng. Nhiều máy khách có thể giữ một biến đồng bộ hóa ở chế độ không dành riêng.
- Khi thực hiện một thao tác acquire, máy khách lấy về phiên bản mới nhất của mục dữ liệu từ tiến trình cuối cùng thực hiện thao tác acquire trên biến đó.

### ĐIỀU KIỆN MÔ HÌNH NHẤT QUÁN ĐI VÀO

- Một thao tác acquire để truy cập vào một biến đồng bộ hóa không được phép thực hiện trong một tiến trình cho đến khi tất cả các cập nhật lên mục dữ liệu trong tiến trình đó được thực hiện.
- Trước khi một truy cập trong chế độ dành riêng của một tiến trình tới một biến đồng bộ hóa được phép thực hiện thì không tiến trình nào khác còn được giữ các biến đồng bộ hóa, trong chế độ không dành riêng thì không cần yêu cầu như vậy.
- Sau khi một truy cập trong chế độ dành riêng lên một biến đồng bộ hóa được thực hiện thì bất kì sự truy cập của tiến trình nào khác trong chế độ không dành riêng lên biến đó cũng không được thực hiện cho đến khi chủ nhân của biến đồng bộ thực hiện xong việc truy cập của mình.

P1:	Acq(Lx)	W(x)a	Acq(Ly)	W(y)b	Rel(Lx)	Rel(Ly)
P2:			Acq(Lx)	R(x)a	R(y)	NIL
P3:				Acq(Ly)	R(y)b	

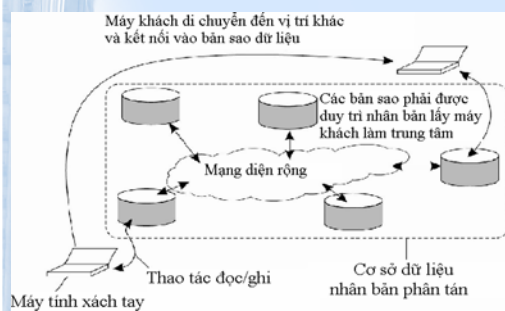
### MÔ HÌNH NHẤT QUÁN LẤY MÁY KHÁCH LÀM TRUNG TÂM

- Các mô hình lấy dữ liệu làm trung tâm nhằm tới cách nhìn nhận tính toàn vẹn dữ liệu toàn bộ hệ thống trong việc lưu trữ dữ liệu.
- Mô hình lấy dữ liệu làm trung tâm chủ yếu giải quyết vấn đề tương tranh và tính tuần tự thực hiện các thao tác.
- Mô hình nhất quán lấy máy khách làm trung tâm bỏ qua các yêu cầu về tính tương tranh:
  - Mô hình nhất quán sau cùng
  - Mô hình nhất quán đọc đều
  - Mô hình nhất quán ghi đều
  - Mô hình nhất quán đọc kết quả ghi
  - Mô hình nhất quán ghi theo sau đọc

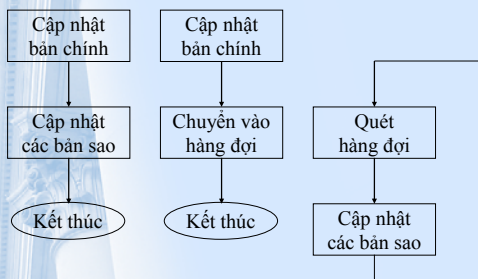
### MÔ HÌNH NHẤT QUÁN SAU CÙNG (Eventual)

- Khi có nhiều bản sao dữ liệu, một yêu cầu đặt ra là sau các thao tác cập nhật thì tất cả các bản sao sau cùng phải giống nhau. Yêu cầu này sẽ được thực hiện tốt nếu mỗi máy khách luôn cập nhật các bản sao.
- Việc cập nhật các bản sao ngay sau khi cập nhật bản chính có thể kéo dài thời gian thực hiện, do đó lập trình viên cần dự đoán thời gian thực hiện mỗi yêu cầu và lựa chọn phương án thích hợp.
- Nếu các máy khách di động thì việc thực hiện yêu cầu trên gặp khó khăn hơn. Phải luôn đảm bảo rằng ngay cả khi máy khách thay đổi về vị trí vật lý thì việc sử dụng các bản sao cũng phải chính xác.

### MÔ HÌNH NHẤT QUÁN SAU CÙNG



### MÔ HÌNH NHẤT QUÁN SAU CÙNG



### MÔ HÌNH NHẤT QUÁN ĐỌC ĐỀU

- Một tiến trình thực hiện thao tác đọc trên một mục dữ liệu thì phải đảm bảo bất kì thao tác đọc nào cũng đều cho cùng một kết quả hay kết quả gần nhất.
- Mô hình nhất quán đọc đều đảm bảo rằng một máy khách sẽ luôn nhìn thấy những dữ liệu mới hơn và không bao giờ phải nhìn thấy những dữ liệu cũ hơn những gì đã đọc trước đó.
- Khi một máy khách thực hiện một thao tác đọc trên một bản sao rồi tiếp theo lại đọc trên một bản sao khác thì bản sao thứ hai ít nhất cũng phải được ghi giống với bản sao đầu tiên.



### MÔ HÌNH NHẤT QUÁN ĐỌC ĐỀU

- Thao tác cập nhật trên L1 sẽ được lan truyền trên L2

L1:  $WS(x_1)$   $R(x_1)$  - - - - -  
 L2:  $WS(x_1; x_2)$   $R(x_2)$

- Không có gì đảm bảo thao tác cập nhật trên L1 sẽ được lan truyền sang L2 và ngược lại.

L1:  $WS(x_1)$   $R(x_1)$  - - - - -  
 L2:  $WS(x_2)$   $R(x_2)$

### MÔ HÌNH NHẤT QUÁN GHI ĐỀU

- Thao tác ghi trên mục dữ liệu  $x$  của một tiến trình phải được hoàn thành trước bất kỳ một thao tác ghi nào khác trên  $x$  bởi cùng một tiến trình.
- Các thao tác ghi lên một mục dữ liệu sẽ được sắp xếp một cách có trật tự.

L1:  $W(x_1)$  - - - - -  
 L2:  $WS(x_1)$  - - - - -  $W(x_2)$   
 L1:  $W(x_1)$  - - - - -  
 L2:  $W(x_2)$

### MÔ HÌNH NHẤT QUÁN ĐỌC KẾT QUẢ GHI

- Người dùng được đảm bảo sẽ luôn được nhìn thấy những kết quả ghi mới nhất.
- Tác động của một thao tác ghi của một tiến trình lên mục dữ liệu  $x$  sẽ luôn được nhìn thấy bởi một thao tác đọc lần lượt trên  $x$  của cùng tiến trình đó.

L1:  $W(x_1)$  - - - - -  
 L2:  $WS(x_1; x_2)$  - - - - -  $R(x_2)$   
 L1:  $W(x_1)$  - - - - -  
 L2:  $WS(x_2)$  - - - - -  $R(x_2)$

### MÔ HÌNH NHẤT QUÁN GHI THEO SAU ĐỌC

- Mô hình nhất quán này ngược với nhất quán đọc kết quả ghi, nó đảm bảo người dùng sẽ luôn thực hiện thao tác ghi lên một phiên bản dữ liệu mà ít nhất cũng phải mới bằng phiên bản cuối cùng của nó.
- Tác động bởi một thao tác ghi của một tiến trình lên mục dữ liệu  $x$  sẽ luôn được nhìn thấy bởi một thao tác đọc liên tiếp lên  $x$  của cùng tiến trình đó.

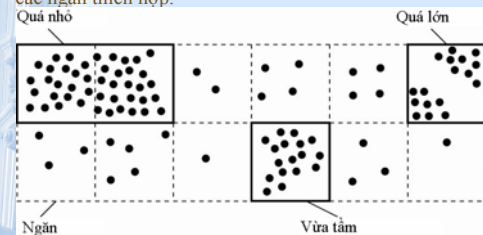
L1:  $WS(x_1)$   $R(x_1)$  - - - - -  
 L2:  $WS(x_1; x_2)$  - - - - -  $W(x_2)$   
 L1:  $WS(x_1)$   $R(x_1)$  - - - - -  
 L2:  $WS(x_2)$  - - - - -  $W(x_2)$

### QUẢN LÝ BẢN SAO

- Việc nhân bản trong hệ thống phân tán cần phải trả lời 5W (When, Where, Who, What, Why) nhằm đảm bảo tính nhất quán của hệ thống.
- Đối với câu hỏi ở đâu (Where) cần phân biệt chi tiết:
  - Máy chủ: Tìm vị trí thích hợp nhất để đặt máy chủ
  - Nội dung: Tìm máy chủ thích hợp nhất để nhân bản nội dung

### SẮP XẾP BẢN SAO-MÁY CHỦ

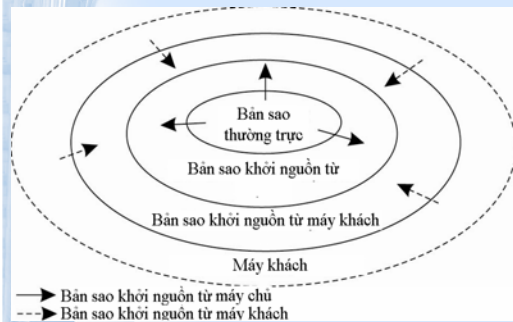
- Xét hình trạng của mạng
- Xét nhu cầu truy nhập nội dung
- Xét cụm máy chủ, sắp xếp các máy chủ trong cụm vào các ngăn thích hợp.



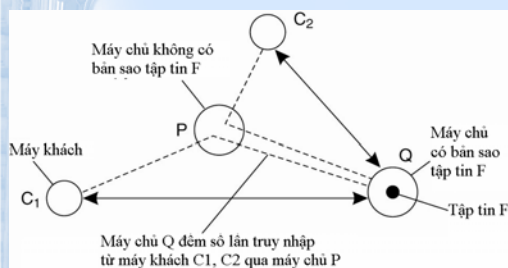
### SẮP XẾP BẢN SAO NỘI DUNG

- Các bản sao thường trực: trong tiến trình hay trên máy luôn có một bản sao. Số lượng các bản sao thường xuyên này rất ít, thường được tập hợp lại thành nhóm các máy, thường là các máy chủ Web hay các máy chủ chứa cơ sở dữ liệu dự phòng.
- Bản sao khởi đầu từ máy chủ: Các bản sao này được sử dụng để làm tăng hiệu năng. Các bản sao này được xếp đặt động dựa vào yêu cầu của máy chủ khác. Một ví dụ điển hình là dịch vụ đặt vị trí trang web sử dụng để xác định vị trí địa lý của các bản sao gần nhất khi cần.
- Các bản sao khởi đầu từ máy khách: Các bản sao này được tạo ra từ yêu cầu của máy khách, chẳng hạn như việc cache dữ liệu của một trình duyệt. Chúng được xếp đặt động dựa vào yêu cầu của máy khách.

### TỔ CHỨC BẢN SAO NỘI DUNG



### BẢN SAO KHỞI NGUỒN TỪ MÁY CHỦ



### PHÂN TÁN NỘI DUNG

- Quản lý bản sao cần phải giải quyết vấn đề lan truyền cập nhật nội dung nhân bản.
- Cần phải cân bằng các yếu tố khác nhau để đạt hiệu năng tốt nhất có thể được:
  - Trạng thái hay thao tác
  - Giao thức kéo hay đẩy
  - Truyền theo điểm-điểm (Unicast) hay theo nhóm (Multicast)

### TRẠNG THÁI HAY THAO TÁC

- Chỉ thông báo là có cập nhật:** Thường dùng trong việc cache dữ liệu. Thông báo về việc mất hiệu lực của một giao thức. Phương pháp này tốt khi tỉ lệ các thao tác đọc so với thao tác ghi nhỏ.
- Truyền dữ liệu cập nhật từ bản sao này tới một bản sao khác:** Thực hiện tốt khi có nhiều thao tác đọc. Ghi lại các thay đổi và tập hợp các cập nhật lại để truyền đi (chỉ truyền đi các thay đổi chứ không truyền cả dữ liệu đã bị thay đổi, vì thế tiết kiệm được băng thông).
- Lan truyền các thao tác cập nhật tới các bản sao khác:** Tốn ít băng thông nhưng đòi hỏi năng lực xử lý cao vì trong nhiều trường hợp các thao tác rất phức tạp.

### CÁC GIAO THỨC KÉO VÀ ĐẨY

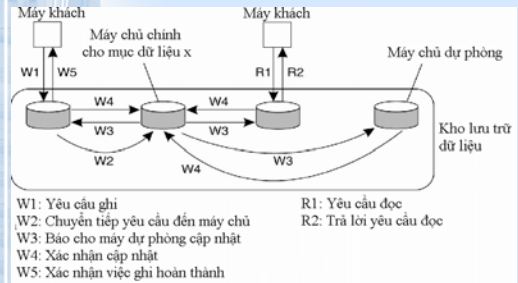
- Giao thức đẩy (Giao thức dựa trên máy chủ):** Mỗi khi có thay đổi, máy chủ sẽ gửi yêu cầu cập nhật đến tất cả các bản sao. Tính nhất quán rất cao.
- Giao thức kéo:** Máy khách hoặc máy chủ gửi yêu cầu đến máy chủ khác để nhận dữ liệu mới nhất (nếu có).

Vấn đề	Đẩy	Kéo
Trạng thái máy chủ	Cần duy trì danh sách các bản sao và cache của máy khách	Không
Thông điệp gửi	Máy chủ gửi yêu cầu cập nhật cho máy khách	Định kỳ quét để lấy yêu cầu cập nhật mới nhất
Thời gian đáp ứng tại máy khách	Ngay lập tức	Thời gian quét

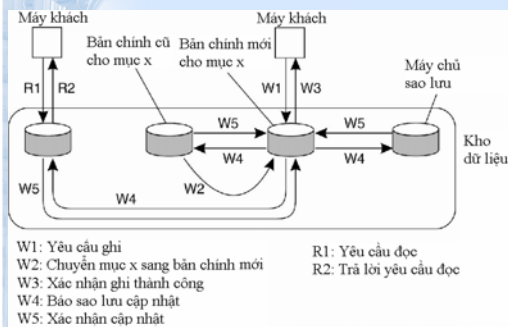
### CÁC GIAO THỨC NHẬT QUẢN

- Các giao thức nhật quản mô tả cách cài đặt các mô hình nhật quản.
- Các giao thức nhật quản dựa trên các giải pháp trong mô hình nhật quản liên tục:
  - Sự chênh lệch giá trị số giữa các bản sao: Giá trị có thể là của một trường dữ liệu nhưng cũng có thể là số lượng các thao tác thay đổi.
  - Sự chênh lệch trạng thái: Thời gian cập nhật cuối cùng
  - Sự chênh lệch thứ tự các thao tác cập nhật: Khả phức tạp, nhất là đối với các thao tác có ROLLBACK

### CÁC GIAO THỨC GHI TỪ XA



### CÁC GIAO THỨC GHI CỤC BỘ



### GIAO THỨC DỰA TRÊN BIỂU QUYẾT

Các thao tác ghi được thực hiện trên một tập nhỏ nhất các bản sao. Khi thực hiện một thao tác đọc, người dùng cũng phải liên hệ với một tập các bản sao để tìm ra phiên bản mới nhất của dữ liệu.

