

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

---

# CƠ SỞ DỮ LIỆU PHÂN TÁN

THIẾT KẾ  
CƠ SỞ DỮ LIỆU PHÂN TÁN  
Phân mảnh dọc

*Ts. Phan Thị Hà*

# *Phân mảnh dọc*

---

## **Định nghĩa**

Phân mảnh dọc quan hệ  $R$  sinh ra các mảnh  $R_1, R_2, \dots, R_r$ , sao cho mỗi mảnh chứa một tập con các thuộc tính của quan hệ  $R$  và khoá của nó.

## **Mục đích**

Phân chia quan hệ  $R$  thành các mảnh nhỏ hơn là để cho nhiều ứng dụng có thể thực hiện chỉ trên một mảnh tối ưu, giảm thiểu thời gian thực hiện ứng dụng. Nâng cao hiệu năng xử lý đồng thời.

## **Tối ưu ?**

Một phân mảnh tối ưu là phân mảnh sinh ra một lược đồ phân mảnh cho phép giảm tối đa thời gian thực thi các ứng dụng chạy trên phân mảnh đó

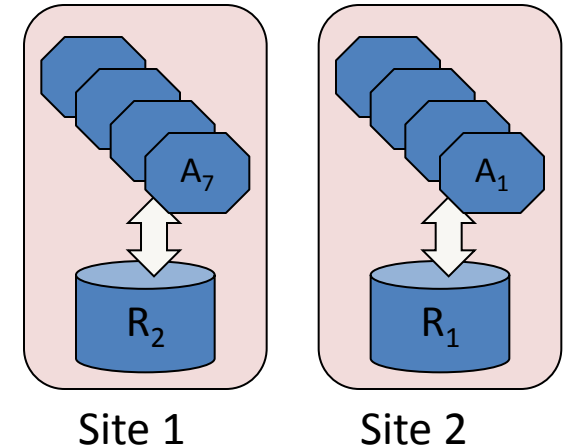
# Phân mảnh dọc

Purpose: Phân mảnh dọc là chúng ta gom những thuộc tính thường được truy xuất chung với nhau vào 1 mảnh.

VD: Hình bên

Purpose Vd: Xác định các phân mảnh R1, R2, các ứng dụng có thể được thực thi chỉ trên 1 phân mảnh

AdvantageVD: Khi nhiều ứng dụng sử dụng R1 và nhiều ứng dụng sử dụng R2 ở các site khác nhau, giảm thiểu thời gian thực hiện ứng dụng. Nâng cao hiệu năng xử lý đồng thời.



Phân mảnh dọc là chúng ta gom những thuộc tính thường được truy xuất chung với nhau vào 1 mảnh.  
Để tiến hành phân mảnh dọc chúng ta cần 1 số thông tin có liên quan đến ứng dụng( các câu truy vấn) và các thuộc tính của quan hệ cần phân hoạch:

- Ma trận sử dụng thuộc tính: biểu diễn mối liên hệ giữa các câu truy vấn và các thuộc tính

- Ma trận lực hút AA: Đo lực hút giữa 2 thuộc tính ( $A_i, A_j$ ).

- Ma trận lực hút tự nhóm: đã gom nhóm các thuộc tính lại với nhau (các thuộc tính có số đo lực hút gần bằng nhau thì được xếp kề nhau)

# Split Approach phân mảnh dọc

1. Ma trận sử dụng thuộc tính ( $A$ )
2. Thu được ma trận lực hút thuộc tính ( $AA$ ) từ ma trận  $A$ , tần số truy cập của mỗi truy vấn  $q_k$  vào cặp thuộc tính  $(A_i, A_j)$  trên mỗi site và tần số truy cập của  $q_k$  vào mỗi site
3. Sử dụng thuật tụ nhóm (BEA) để nhóm các thuộc tính xuất hiện cùng nhau dựa vào ma trận lực hút thuộc tính. Thuật toán này sản xuất ra ma trận lực hút tụ nhóm. ( $CA$ )
4. Sử dụng thuật toán Phân mảnh dọc tách các thuộc tính such that set of attributes are accessed solely or for the most part by distinct set of applications.

## ***Ma trận giá trị sử dụng thuộc tính***

---

- ❑  $R(A_1, A_2, \dots, A_n)$  quan hệ toàn cục
- ❑  $Q = \{q_1, q_2, \dots, q_m\}$  tập các ứng dụng
- ❑ Ma trận giá trị sử dụng thuộc tính định nghĩa như sau:

$$A = (\text{use}(q_i, A_j))_{m \times n}$$

$$\text{use}(q_i, A_j) = \begin{cases} 1 & \text{Nếu } q_i \text{ tham chiếu đến thuộc tính } A_j \\ 0 & \text{Ngược lại} \end{cases}$$

$$i = 1..m \text{ và } j = 1..n$$

$$n = |\Omega| \text{ và } m = |Q|$$

## ***Mã trận giá trị sử dụng thuộc tính***

---

**A** =

|                      | <b>A1</b>                    | <b>A2</b>                    | <b>.....</b> | <b>A<sub>n</sub></b>                    |
|----------------------|------------------------------|------------------------------|--------------|---|
| <b>q1</b>            | <b>Use(q1,A1)</b>            | <b>Use(q1,A2)</b>            |              | <b>Use(q1,A<sub>n</sub>)</b>            |
| <b>q2</b>            | <b>Use(q2,A1)</b>            | <b>Use(q2,A2)</b>            |              | <b>Use(q2,A<sub>n</sub>)</b>            |
| <b>....</b>          | <b>.....</b>                 | <b>.....</b>                 | <b>...</b>   | <b>....</b>                             |
| <b>q<sub>m</sub></b> | <b>Use(q<sub>m</sub>,A1)</b> | <b>Use(q<sub>m</sub>,A2)</b> |              | <b>Use(q<sub>m</sub>,A<sub>n</sub>)</b> |

## *Ví dụ ma trận giá trị sử dụng thuộc tính*

---

Quan hệ: PROJ (PNO, PNAME, BUDGET, LOC)

*Tập các ứng dụng:*

q1: Kinh phí của dự án khi biết mã dự án

SELECT BUDGET FROM PROJ WHERE PNO = Value

q2: Tên và kinh phí của tất các dự án

SELECT PNAME, BUDGET FROM PROJ

q3: Tìm tên các dự án khi biết thành phố

SELECT PNAME FROM PROJ WHERE LOC = Value

q4: Tổng kinh phí của các dự án tại mỗi thành phố

SELECT SUM(BUDGET) FROM PROJ WHERE LOC = Value



## *Ví dụ ma trận giá trị sử dụng thuộc tính*

---

Ký hiệu: A1= PNO, A2=PNAME, A3=BUDGET, A4=LOC

q1: SELECT A3 FROM PROJ WHERE A1= Value

q2: SELECT A2, A3 FROM PROJ

q3: SELECT A2 FROM PROJ WHERE A4 = Value

q4: SELECT SUM(A3) FROM PROJ WHERE A4= Value

$$\mathbf{A} = \begin{matrix} & \begin{matrix} A_1 & A_2 & A_3 & A_4 \end{matrix} \\ \begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

## Trọng số lực hút (Attribute Affinit Measure)

- ☐  ~~$R(A_1, A_2, \dots, A_n)$  quan hệ toàn cục~~
- ☐  $Q = \{q_1, q_2, \dots, q_m\}$  tập các ứng dụng
- ☐ Bảng tần số ứng dụng trên các site:  $S = \{S_1, S_2, \dots, S_t\}$
- ☐ Khi đó  $AA = (\text{aff}(A_i, A_j))_{n \times n}$  Ma trận lực hút
- ☐  $\text{aff}(A_i, A_j)$ : Trọng số lực hút

$$\text{aff}(A_i, A_j) = \sum_{k: [(use(q_k, A_i) \wedge (use(q_k, A_j)) = 1 \forall S_l]} \sum_{l} \text{ref}_l(q_k) \text{acc}_l(q_k)$$

*Trong đó:*

- ☐  $\text{ref}_l(q_k)$  là số truy suất của trên  $(A_i, A_j)$  cho  $q_k$  tại vị trí  $S_l$
- ☐  $\text{acc}_l(q_k)$  là tần số truy suất của  $q_k$  tại vị trí  $S_l$

## ***$Ma\ tr\grave{a}n\ l\grave{y}c\ h\grave{u}t\ AA(Attribute\ Affinity\ Matrix)$***

---

**$AA =$**

|                      | <b>A1</b>                       | <b>A2</b>                       | <b>....</b> | <b>An</b>                        |
|----------------------|---------------------------------|---------------------------------|-------------|----------------------------------|
| <b>A1</b>            | <b><math>aff(A1,A1)</math></b>  | <b><math>aff(A1,A2)</math></b>  |             | <b><math>aff(A1,A_n)</math></b>  |
| <b>A2</b>            | <b><math>aff(A2,A1)</math></b>  | <b><math>aff(A2,A2)</math></b>  |             | <b><math>aff(A2,A_n)</math></b>  |
| <b>....</b>          | <b>.....</b>                    | <b>.....</b>                    | <b>...</b>  | <b>....</b>                      |
| <b>A<sub>n</sub></b> | <b><math>aff(A_n,A1)</math></b> | <b><math>aff(A_n,A2)</math></b> |             | <b><math>aff(A_n,A_n)</math></b> |

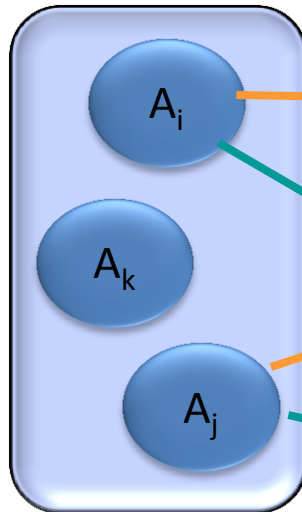
$$aff(A_i, A_j) = \sum_{\forall k, use(q_k, A_i)=1 \wedge use(q_k, A_j)=1} \sum_{\forall s} ref_s(q_k) \cdot acc_s(q_k)$$

*For each query  $q_k$  that uses both  $A_i$  and  $A_j$*

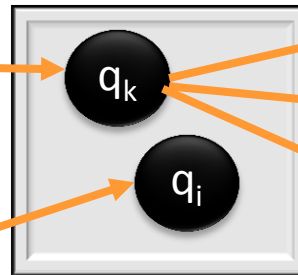
*Popularity of such  $A_i$ - $A_j$  pair at all sites*

Popularity of using  $A_i$  and  $A_j$  together

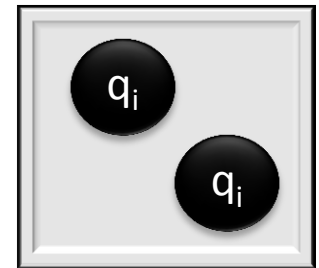
Relation  $R$



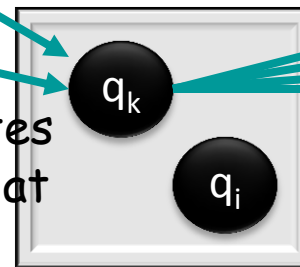
Site  $m$



Site  $n$

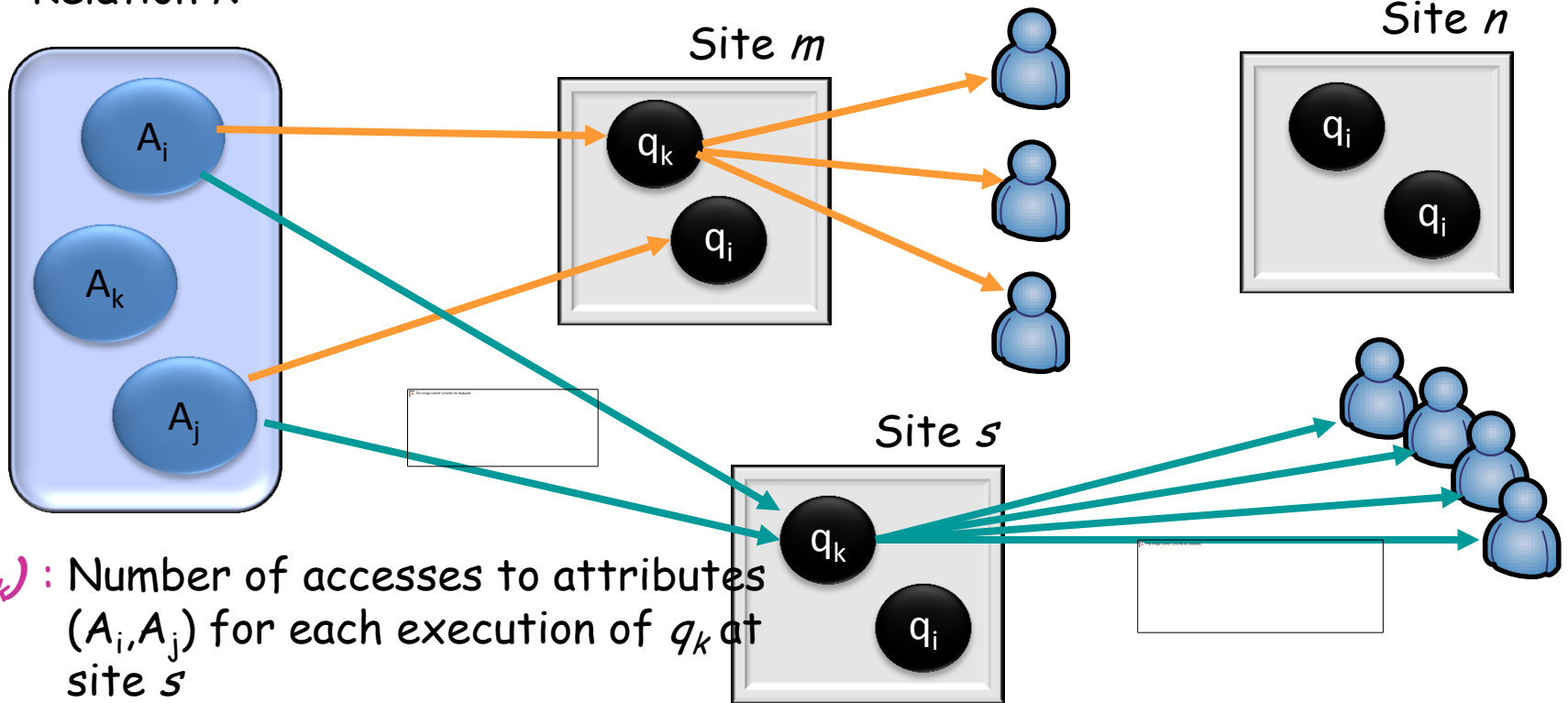


Site  $s$



$ref_s(q_k)$  : Number of accesses to attributes  $(A_i, A_j)$  for each execution of  $q_k$  at site  $s$

$acc_s(q_k)$  : Application access frequency of  $q_k$  at site  $s$ .



## *Ví dụ ma trận lực hút AA*

---

- ❑ Giả sử  $\text{ref}_i(q_k) = 1$  cho tất cả  $q_k$  và  $S_i$
- ❑ Giả sử tần số các ứng dụng trên các Site là:

### Site1

$$\text{acc}_1(q_1)=15$$

$$\text{acc}_1(q_2)=5$$

$$\text{acc}_1(q_3)=25$$

$$\text{acc}_1(q_4)=3$$

### Site2

$$\text{acc}_2(q_1)=20$$

$$\text{acc}_2(q_2)=0$$

$$\text{acc}_2(q_3)=25$$

$$\text{acc}_2(q_4)=0$$

### Site3

$$\text{acc}_3(q_1)=10$$

$$\text{acc}_3(q_2)=0$$

$$\text{acc}_3(q_3)=25$$

$$\text{acc}_3(q_4)=0$$

## Ví dụ ma trận lực hút AA

### Site1

$$acc_1(q_1)=15$$

$$acc_1(q_2)=5$$

$$acc_1(q_3)=25$$

$$acc_1(q_4)=3$$

### Site2

$$acc_2(q_1)=20$$

$$acc_2(q_2)=0$$

$$acc_2(q_3)=25$$

$$acc_2(q_4)=0$$

### Site3

$$acc_3(q_1)=10$$

$$acc_3(q_2)=0$$

$$acc_3(q_3)=25$$

$$acc_3(q_4)=0$$

$$aff(A_1, A_3) = \sum_{k=1}^4 \sum_{l=1}^3 acc_l(q_k) = acc_1(q_1) + acc_2(q_1) + acc_3(q_1) = 45$$

$$\mathbf{A} = \begin{matrix} & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 \\ \mathbf{q}_1 & \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{q}_2 & \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \\ \mathbf{q}_3 & \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \\ \mathbf{q}_4 & \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$\mathbf{AA} =$$

|                | $\mathbf{A}_1$ | $\mathbf{A}_2$ | $\mathbf{A}_3$ | $\mathbf{A}_4$ |
|----------------|----------------|----------------|----------------|----------------|
| $\mathbf{A}_1$ | 45             | 0              | 45             | 0              |
| $\mathbf{A}_2$ | 0              | 80             | 5              | 75             |
| $\mathbf{A}_3$ | 45             | 5              | 53             | 3              |
| $\mathbf{A}_4$ | 0              | 75             | 3              | 78             |

## *Ví dụ ma trận lực hút AA*

---

AA =

|                      | <b>A<sub>1</sub></b> | <b>A<sub>2</sub></b> | <b>A<sub>3</sub></b> | <b>A<sub>4</sub></b> |
|----------------------|----------------------|----------------------|----------------------|----------------------|
| <b>A<sub>1</sub></b> | 45                   | 0                    | 45                   | 0                    |
| <b>A<sub>2</sub></b> | 0                    | 80                   | 5                    | 75                   |
| <b>A<sub>3</sub></b> | 45                   | 5                    | 53                   | 3                    |
| <b>A<sub>4</sub></b> | 0                    | 75                   | 3                    | 78                   |

# Ma trận lực hút tụ nhóm (CA)

- Sử dụng thuật tụ nhóm (BEA) để nhóm các thuộc tính xuất hiện cùng nhau dựa vào ma trận lực hút thuộc tính AA.
- Thuật toán hoán vị các hàng và các cột của ma trận AA, sao cho số đo lực hút chung AM là lớn nhất

$$AM = \sum_i \sum_j (\text{affinity of } A_i \text{ and } A_j \text{ with their neighbors})$$



## ***Thuật toán tụ nhóm BEA (Bond Energy Algorithm)***

---

Nhóm các thuộc tính của quan hệ toàn cục bằng cách hoán vị các hàng và các cột của ma trận AA, sao cho số đo hấp dẫn **cont()** là lớn nhất. Kết quả sẽ là một ma trận tụ hấp dẫn CA (Cluster Affinity). Thuật toán

- **Input:** Ma trận AA
- **Output:** Ma trận quan hệ phân cụm CA

*Bước 1. Khởi tạo:* Đặt cột 1 và 2 của AA vào cột 1&2 trong CA.

*Bước 2:* Giả sử có  $i$  cột đã được đặt vào CA. Lấy lần lượt một trong  $(n-i)$  cột còn lại của AA, đặt vào cột thứ  $(i+1)$  của CA, sao cho số đo AM tại vị trí đó là lớn nhất. Lặp lại bước 2 cho đến hết

*Bước 3:* Sắp thứ tự hàng theo thứ tự cột

- Số đo đóng góp của thuộc tính  $A_k$  khi đặt vào  $A_i$  và  $A_j$  (cont)

Điều kiện biên:  $A_k, A_i, A_j ; A_i, A_k, A_j ; A_i, A_j, A_k$

Xếp  $A_k$  vào vị trí ngoài cùng bên trái: Thêm cột  $A_0$

Xếp  $A_k$  vào vị trí ngoài cùng bên phải: Thêm cột  $A_n$

Các cột  $A_0$  và  $A_n$  có các phần tử = 0 trong ma trận bù thuộc tính

- $$AM = \sum_{i=1}^n \sum_{j=1}^n aff(A_i, A_j) [aff(A_i, A_{j-1}) + aff(A_i, A_{j+1}) + aff(A_{i-1}, A_j) + aff(A_{i+1}, A_j)]$$

where

$$aff(A_0, A_j) = aff(A_i, A_0) = aff(A_{n+1}, A_j) = aff(A_i, A_{n+1}) = 0$$

- Vì ma trận AA đối xứng, nên khi tính AM ta có thể giảm chức năng mục tiêu của AM để

$$AM = \sum_{i=1}^n \sum_{j=1}^n aff(A_i, A_j) [aff(A_i, A_{j-1}) + aff(A_i, A_{j+1})]$$

which can be rewritten as

$$\begin{aligned} AM &= \sum_{i=1}^n \sum_{j=1}^n [aff(A_i, A_j)aff(A_i, A_{j-1}) + aff(A_i, A_j)aff(A_i, A_{j+1})] \\ &= \sum_{j=1}^n \left[ \sum_{i=1}^n aff(A_i, A_j)aff(A_i, A_{j-1}) + \sum_{i=1}^n aff(A_i, A_j)aff(A_i, A_{j+1}) \right] \end{aligned}$$

Let us define the *bond* between two attributes  $A_x$  and  $A_y$  as

$$bond(A_x, A_y) = \sum_{z=1}^n aff(A_z, A_x)aff(A_z, A_y)$$

Then  $AM$  can be written as

$$AM = \sum_{j=1}^n [bond(A_j, A_{j-1}) + bond(A_j, A_{j+1})]$$

$$\underbrace{A_1 A_2 \dots A_{i-1} A_i}_{AM'} A_j \underbrace{A_{j+1} \dots A_n}_{AM''}$$

The global affinity measure for these attributes can be written as

$$AM_{old} = AM' + AM'' + \text{bond}(A_{i-1}, A_i) + \text{bond}(A_i, A_j) + \text{bond}(A_j, A_{j+1})$$

$$AM_{new} = AM' + AM'' +$$

$$\text{bond}(A_{i-1}, A_i) + \text{bond}(A_i, A_k) + \text{bond}(A_k, A_j) + \text{bond}(A_j, A_{j+1})$$

=> Vậy đóng góp cho AM chung khi đặt  $A_k$  vào giữa giữa  $A_i, A_j$ ,

$$AM_{new} - AM_{old} = 2\text{bond}(A_k, A_i) + \text{bond}(A_k, A_j) - \text{bond}(A_i, A_j)$$

$$\begin{aligned} cont(A_i, A_k, A_j) &= AM_{new} - AM_{old} \\ &= 2\text{bond}(A_i, A_k) + \text{bond}(A_k, A_j) - \text{bond}(A_i, A_j) \end{aligned}$$

=> chọn vị trí để đặt  $A_k$  để AM max tương đương với  $Cont(A_i, A_k, A_j) \max > 0$

*vd*

$$\text{cont}(A_1, A_4, A_2) = 2[\text{bond}(A_1, A_4) + \text{bond}(A_4, A_2) - \text{bond}(A_1, A_2)]$$

$$\begin{aligned} \text{bond}(A_1, A_4) = & \text{aff}(A1, A1) * \text{aff}(A1, A4) + \\ & \text{aff}(A2, A1) * \text{aff}(A2, A4) + \\ & \text{aff}(A3, A1) * \text{aff}(A3, A4) + \\ & \text{aff}(A4, A1) * \text{aff}(A4, A4) \end{aligned}$$

$$\text{bond}(A_1, A_4) = 135$$

$$\text{bond}(A_4, A_2) = 11865$$

$$\text{bond}(A_1, A_2) = 225$$

|    | A1 | A2 | A3 | A4 |
|----|----|----|----|----|
| A1 | 45 | 0  | 45 | 0  |
| A2 | 0  | 80 | 5  | 75 |
| A3 | 45 | 5  | 53 | 3  |
| A4 | 0  | 75 | 3  | 78 |

$$\text{cont}(A_1, A_4, A_2) = 2 * 135 + 2 * 11865 - 2 * 225 = 23550$$

# *Giải mã Thuật toán tụ nhóm BEA (Bond Energy*

---

## **Algorithm 3.3:** BEA Algorithm

---

**Input:** *AA*: attribute affinity matrix

**Output:** *CA*: clustered affinity matrix

**begin**

    {initialize; remember that *AA* is an  $n \times n$  matrix}

$CA(\bullet, 1) \leftarrow AA(\bullet, 1)$  ;

$CA(\bullet, 2) \leftarrow AA(\bullet, 2)$  ;

$index \leftarrow 3$  ;

**while**  $index \leq n$  **do**           {choose the “best” location for attribute  $AA_{index}$ }

**for**  $i$  from 1 to  $index - 1$  by 1 **do** calculate  $cont(A_{i-1}, A_{index}, A_i)$  ;

        calculate  $cont(A_{index-1}, A_{index}, A_{index+1})$  ;           {boundary condition}

$loc \leftarrow$  placement given by maximum  $cont$  value ;

**for**  $j$  from  $index$  to  $loc$  by  $-1$  **do**

$CA(\bullet, j) \leftarrow CA(\bullet, j - 1)$                            {shuffle the two matrices}

$CA(\bullet, loc) \leftarrow AA(\bullet, index)$  ;

$index \leftarrow index + 1$

    order the rows according to the relative ordering of columns

**end**

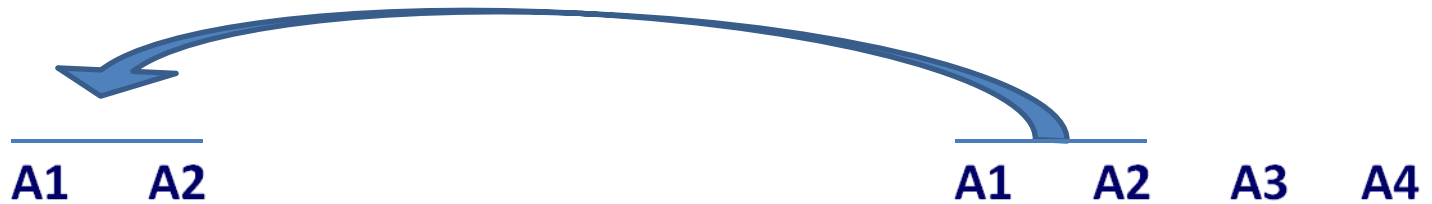
## Ví dụ

---

Chép cột 1 và cột 2 ma trận AA vào ma trận CA

$$(1) CA(*,1) \leftarrow AA(*,1)$$

$$(2) CA(*,2) \leftarrow AA(*,2)$$



**CA =**

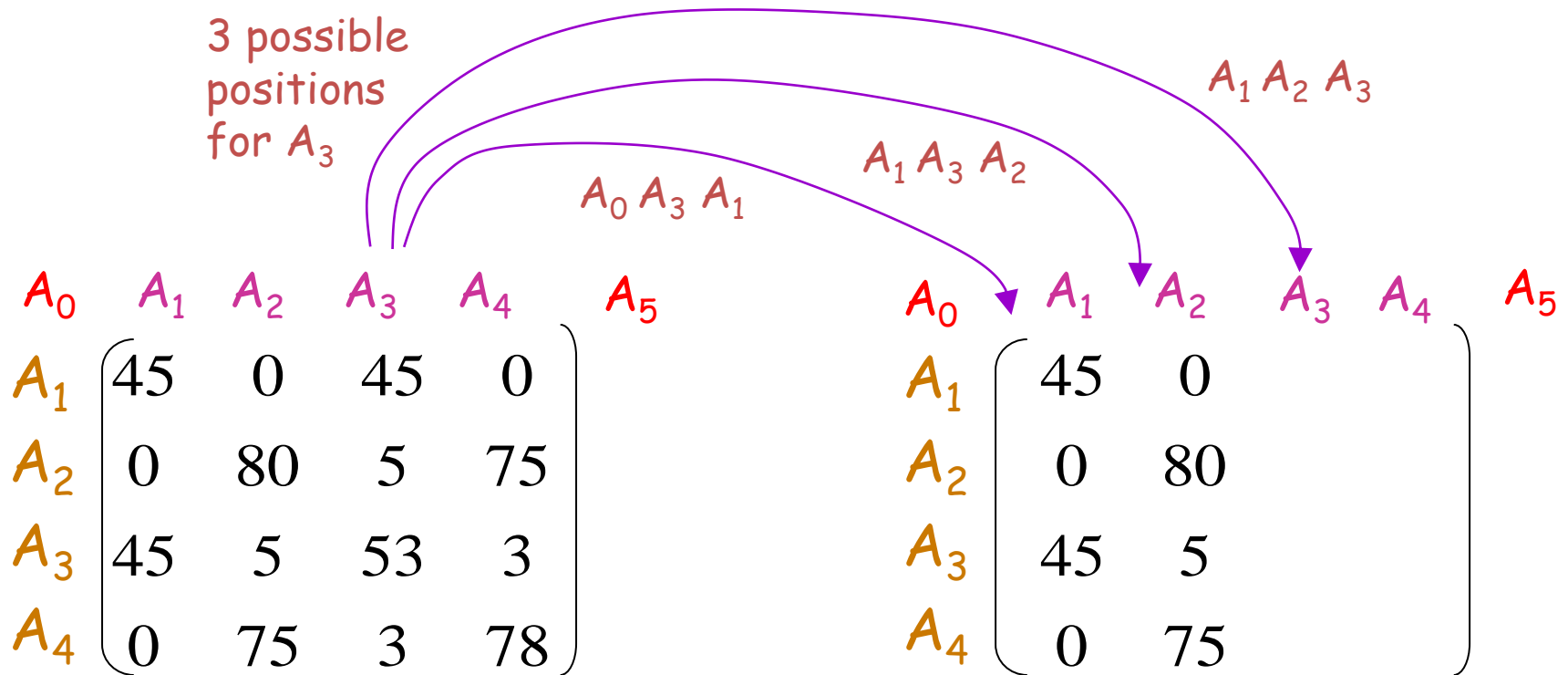
|    | A1 | A2 |  |  |
|----|----|----|--|--|
| A1 | 45 | 0  |  |  |
| A2 | 0  | 80 |  |  |
| A3 | 45 | 5  |  |  |
| A4 | 0  | 75 |  |  |

**AA =**

|    | A1 | A2 | A3 | A4 |
|----|----|----|----|----|
| A1 | 45 | 0  | 45 | 0  |
| A2 | 0  | 80 | 5  | 75 |
| A3 | 45 | 5  | 53 | 3  |
| A4 | 0  | 75 | 3  | 78 |

# Clustered Affinity Matrix

## Step 2: Determine Location for $A_3$



Attribute Affinity Matrix (AA)

Clustered Affinity Matrix (CA)



## *Ví dụ*

---

*index=3*

**While** *index*  $\leq n$  **do**

*index*  $\leq 4$  {thỏa mãn}

**For** *i* **from** 1 **to** *index* – 1 **by** 1 **do**

Tính  $\text{cont}(A_{i-1}, A_{\text{index}}, A_i)$

*i*=1    thứ tự ( 0-3-1):  $\text{cont}(A_0, A_3, A_1) = 8820$

*i*=2    thứ tự (1-3-2):  $\text{cont}(A_1, A_3, A_2) = 10150$

**End – for**

Điều kiện biên, thứ tự (2-3-4):  $\text{cont}(A_2, A_3, A_4) = 1780$


*loc* =2 thứ tự (1-3-2) có  $\text{cont} = 10150$  lớn nhất

**For** *j* **from** *index* **to** *Loc* **by** – 1 **do** {xáo trộn hai ma trận}

$CA(*, j) := AA(*, j-1)$

## Ví dụ

---



**CA =**

|    | A1 | A3 | A2 |  |
|----|----|----|----|--|
| A1 | 45 | 45 | 0  |  |
| A2 | 0  | 5  | 80 |  |
| A3 | 45 | 53 | 5  |  |
| A4 | 0  | 3  | 75 |  |

**AA =**

|    | A1 | A2 | A3 | A4 |
|----|----|----|----|----|
| A1 | 45 | 0  | 45 | 0  |
| A2 | 0  | 80 | 5  | 75 |
| A3 | 45 | 5  | 53 | 3  |
| A4 | 0  | 75 | 3  | 78 |

Đặt  $A_3$  giữa  $A_1$  và  $A_2$

---

*index=4*

**While** *index*  $\leq n$  **do**

*index*  $\leq 4$  {thỏa mãn}

**For** *i* **from** 1 **to** *index* - 1 **by** 1 **do**

Tính  $\text{cont}(A_{i-1}, A_{\text{index}}, A_i)$

*i*=1    thứ tự (0-4-1):  $\text{cont}(A_0, A_4, A_1) = 270$

*i*=2    thứ tự (1-4-2):  $\text{cont}(A_1, A_4, A_3) = - 5208$

*i*=3    thứ tự (2-4-3):  $\text{cont}(A_3, A_4, A_2) = 23698$

**End – for**

Điều kiện biên, thứ tự (3-4-5):  $\text{cont}(A_3, A_4, A_5) = 23730$

loc =4 thứ tự (3-4-5) có  $\text{cont} = 23730$  lớn nhất

**CA =**

|    | A1 | A3 | A2 | <u>A4</u> |
|----|----|----|----|-----------|
| A1 | 45 | 45 | 0  | 0         |
| A2 | 0  | 5  | 80 | 75        |
| A3 | 45 | 53 | 5  | 3         |
| A4 | 0  | 3  | 75 | 78        |

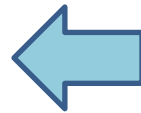
**AA =**

|    | A1 | A2 | A3 | <u>A4</u> |
|----|----|----|----|-----------|
| A1 | 45 | 0  | 45 | 0         |
| A2 | 0  | 80 | 5  | 75        |
| A3 | 45 | 5  | 53 | 3         |
| A4 | 0  | 75 | 3  | 78        |

Đặt  $A_4$  bên phải  $A_2$

**CA =**

|    | A1 | A3 | A2 | A4 |
|----|----|----|----|----|
| A1 | 45 | 45 | 0  | 0  |
| A3 | 45 | 53 | 5  | 3  |
| A2 | 0  | 5  | 80 | 75 |
| A4 | 0  | 3  | 75 | 78 |



**CA =**

|    | A1 | A3 | A2 | A4 |
|----|----|----|----|----|
| A1 | 45 | 45 | 0  | 0  |
| A2 | 0  | 5  | 80 | 75 |
| A3 | 45 | 53 | 5  | 3  |
| A4 | 0  | 3  | 75 | 78 |

## *Chú ý Thuật toán tụ nhóm*

---

- ❑ Độ đo cầu nối giữa hai thuộc tính được tính là tổng của tích 2 phần tử cùng hàng của hai cột. Vì ma trận  $AA$  đối xứng, có thể thực hiện tương tự theo hàng.
- ❑ Trong bước khởi gán, cột 1 và 2 được đặt vào vị trí 1&2 trong  $CA$ , vì  $A_2$  có thể đặt ở bên trái hoặc phải của  $A_1$ .
- ❑ Nếu  $A_j$  là thuộc tính tận trái trong ma trận  $CA$ , kiểm tra đóng góp khi đặt thuộc tính  $A_k$  vào bên trái của  $A_j$ , khi đó  $\text{bond}(A_0, A_k) = \text{bond}(A_0, A_j) = 0$ ,
- ❑ Nếu  $A_j$  là thuộc tính tận phải đã được đặt trong ma trận  $CA$  và đang kiểm tra đóng góp khi đặt thuộc tính  $A_k$  vào bên phải của  $A_j$ , Khi đó  $\text{bond}(A_j, A_{k+1}) = \text{bond}(A_k, A_{k+1}) = 0$ .

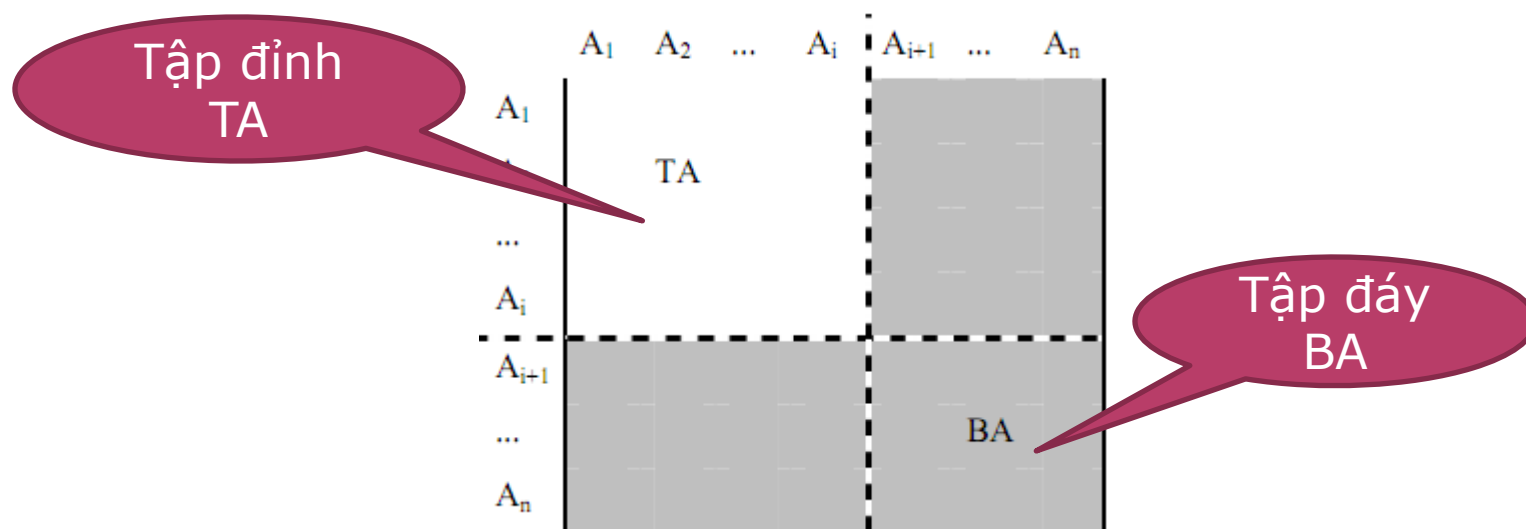
# *Thuật toán phân mảnh dọc*

- Làm thế nào để chia một tập các thuộc tính đã được phân cụm  $\{A_1, A_2, \dots, A_n\}$  thành hai (hoặc nhiều hơn) tập  $\{A_1, A_2, \dots, A_i\}$  và  $\{A_i, \dots, A_n\}$  sao cho không có (hoặc tối thiểu) ứng dụng nào truy cập cả hai (hoặc nhiều hơn một) tập

# Thuật toán phân mảnh dọc

Xét ma trận lực hút tự nhóm (tự lực) CA

- ❑  $TA = \{A_1, A_2, \dots, A_i\}$  ở góc trái cao nhất gọi là tập đỉnh (Top)
- ❑  $BA = \{A_{i+1}, A_{i+2}, \dots, A_n\}$  ở góc phải thấp nhất gọi là tập đáy (Bottom)





## *Thuật toán phân mảnh dọc*

---

| Ký hiệu   | Ý nghĩa   |
|---|---|
| $Q = \{q_1, q_2, \dots, q_n\}$                    | Tập các ứng dụng.                                     |
| $AQ(q_i) = \{A_j \mid \text{use}(q_i, A_j) = 1\}$ | Tập các thuộc tính được truy xuất bởi ứng dụng $q_i$  |
| $TQ = \{q_i \mid AQ(q_i) \subseteq TA\}$          | Tập các ứng dụng chỉ truy xuất trên các thuộc tính TA |
| $BQ = \{q_i \mid AQ(q_i) \subseteq BA\}$          | Tập các ứng dụng chỉ truy xuất trên các thuộc tính BA |
| $OQ = Q - \{TQ \cup BQ\}$                         | Tập các ứng dụng truy xuất trên cả BA và TA           |

# Thuật toán phân mảnh dọc

---

## Ký hiệu

## Ý nghĩa

$$CQ = \sum_{q_i \in \Omega} \sum_{\forall S_j} ref_j(q_i) acc_j(q_i)$$

Tổng chi phí truy xuất của tất cả các ứng dụng trên tất cả các vị trí

$$CTQ = \sum_{qi \in TQ} \sum_{\forall S_j} ref_j(q_i).acc_j(q_i)$$

Tổng số các truy cập đến các thuộc tính bởi các ứng dụng chỉ truy cập TA

$$CBQ = \sum_{qi \in BQ} \sum_{\forall S_j} ref_j(q_i).acc_j(q_i)$$

Tổng số các truy cập đến các thuộc tính bởi các ứng dụng chỉ truy cập BA

$$COQ = \sum_{qi \in OQ} \sum_{\forall S_j} ref_j(q_i).acc_j(q_i)$$

Tổng số các truy cập đến các thuộc tính bởi ứng dụng truy cập cả TA và BA

# *Thuật toán phân mảnh dọc*

---

Bài toán tối ưu hóa phân mảnh chính là bài toán xác định định một điểm :  $1 \leq z \leq n$  sao cho :

$$z = \text{CTQ}^* \text{CBQ} - \text{COQ}^2 \text{ là lớn nhất}$$

# Giả mã Thuật toán phân mảnh dọc

---

## Algorithm 3.4: PARTITION Algorithm

---

**Input:**  $CA$ : clustered affinity matrix;  $R$ : relation;  $ref$ : attribute usage matrix;  
 $acc$ : access frequency matrix

**Output:**  $F$ : set of fragments

**begin**

    {determine the  $z$  value for the first column}

    {the subscripts in the cost equations indicate the split point}

    calculate  $CTQ_{n-1}$  ;

    calculate  $CBQ_{n-1}$  ;

    calculate  $COQ_{n-1}$  ;

$best \leftarrow CTQ_{n-1} * CBQ_{n-1} - (COQ_{n-1})^2$  ;

**repeat**

        {determine the best partitioning}

**for**  $i$  from  $n - 2$  to 1 **by**  $-1$  **do**

            calculate  $CTQ_i$  ;

            calculate  $CBQ_i$  ;

            calculate  $COQ_i$  ;

$z \leftarrow CTQ_i * CBQ_i - COQ_i^2$  ;

**if**  $z > best$  **then**  $best \leftarrow z$                       {record the split point within shift}

        call  $SHIFT(CA)$

**until** no more  $SHIFT$  is possible ;

    reconstruct the matrix according to the shift position ;

$R_1 \leftarrow \Pi_{TA}(R) \cup K$  ;                      { $K$  is the set of primary key attributes of  $R$ }

$R_2 \leftarrow \Pi_{BA}(R) \cup K$  ;

$F \leftarrow \{R_1, R_2\}$

**end**

---

# vd Thuật toán phân mảnh dọc

**A** =

|       | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|-------|-------|-------|-------|-------|
| $q_1$ | 1     | 0     | 1     | 0     |
| $q_2$ | 0     | 1     | 1     | 0     |
| $q_3$ | 0     | 1     | 0     | 1     |
| $q_4$ | 0     | 0     | 1     | 1     |
|       | A1    | A3    | A2    | A4    |

**CA** =

| A1 | 45 | 45 | 0  | 0  |
|----|----|----|----|----|
| A3 | 45 | 53 | 5  | 3  |
| A2 | 0  | 5  | 80 | 75 |
| A4 | 0  | 3  | 75 | 78 |

## Vị trí 1:

$AQ(q_1) = \{A_1, A_3\}$     $AQ(q_2) = \{A_2, A_3\}$ ,    $AQ(q_3) = \{A_2, A_4\}$ ,  
 $AQ(q_4) = \{A_3, A_4\}$

$TA = \{A_1\} \Rightarrow TQ = \{q_i \mid AQ(q_i) \subseteq TA\} = \{\}$ ,

$BA = \{A_3, A_2, A_4\} \Rightarrow BQ = \{q_i \mid AQ(q_i) \subseteq BA\} = \{q_2, q_3, q_4\}$

$OQ = \{q_i \mid AQ(q_i) \subseteq TA\} = \{q_1\}$ ,  $AQ(q_i) \subseteq BA\} = \{\}$

CTQ = 0;

$$C_{xx} = \sum_{q_i \in xx} \sum_{vs_j} ref_i(q_i) \times acc_j(q_i)$$

$CBQ = acc_1(q_2) + acc_2(q_2) + acc_3(q_2) +$   
 $acc_1(q_3) + acc_2(q_3) + acc_3(q_3) +$   
 $acc_1(q_4) + acc_2(q_4) + acc_3(q_4) = 83$

$COQ = acc_1(q_1) + acc_2(q_1) + acc_3(q_1) = 45$

$$Z = CTQ * CBQ - COQ^2 = -2025$$

### Site1

$$acc_1(q_1) = 15$$

$$acc_1(q_2) = 5$$

$$acc_1(q_3) = 25$$

$$acc_1(q_4) = 3$$

### Site2

$$acc_2(q_1) = 20$$

$$acc_2(q_2) = 0$$

$$acc_2(q_3) = 25$$

$$acc_2(q_4) = 0$$

### Site3

$$acc_3(q_1) = 10$$

$$acc_3(q_2) = 0$$

$$acc_3(q_3) = 25$$

$$acc_3(q_4) = 0$$

# *vdThuật toán phân mảnh dọc*

## Vị trí 2:

**CA =**

|    | A1 | A3 | A2 | A4 |
|----|----|----|----|----|
| A1 | 45 | 45 | 0  | 0  |
| A3 | 45 | 53 | 5  | 3  |
| A2 | 0  | 5  | 80 | 75 |
| A4 | 0  | 3  | 75 | 78 |

|                | A <sub>1</sub> | A <sub>2</sub> | A <sub>3</sub> | A <sub>4</sub> |
|----------------|----------------|----------------|----------------|----------------|
| q <sub>1</sub> | 1              | 0              | 1              | 0              |
| q <sub>2</sub> | 0              | 1              | 1              | 0              |
| q <sub>3</sub> | 0              | 1              | 0              | 1              |
| q <sub>4</sub> | 0              | 0              | 1              | 1              |

$$TA = \{A_1, A_3\}, TQ = \{q_1\},$$

$$BA = \{A_2, A_4\}, BQ = \{q_3\},$$

$$OQ = \{q_2, q_4\}$$

$$CTQ_2 = acc_1(q_1) + acc_2(q_1) + acc_3(q_1) = 45$$

$$CBQ_2 = acc_1(q_3) + acc_2(q_3) + acc_3(q_3) = 75$$

$$COQ_2 = acc_1(q_2) + acc_2(q_2) + acc_3(q_2) + acc_1(q_4) + acc_2(q_4) + acc_3(q_4) = 8$$

$$Z = CTQ * CBQ - COQ^2 = 3311$$

### Site1

$$acc_1(q_1)=15$$

$$acc_1(q_2)=5$$

$$acc_1(q_3)=25$$

$$acc_1(q_4)=3$$

### Site2

$$acc_2(q_1)=20$$

$$acc_2(q_2)=0$$

$$acc_2(q_3)=25$$

$$acc_2(q_4)=0$$

### Site3

$$acc_3(q_1)=10$$

$$acc_3(q_2)=0$$

$$acc_3(q_3)=25$$

$$acc_3(q_4)=0$$

## Vd Thuật toán phân mảnh dọc

CA =

|    | A1 | A3 | A2 | A4 |
|----|----|----|----|----|
| A1 | 45 | 45 | 0  | 0  |
| A3 | 45 | 53 | 5  | 3  |
| A2 | 0  | 5  | 80 | 75 |
| A4 | 0  | 3  | 75 | 78 |

|                | A <sub>1</sub> | A <sub>2</sub> | A <sub>3</sub> | A <sub>4</sub> |
|----------------|----------------|----------------|----------------|----------------|
| q <sub>1</sub> | 1              | 0              | 1              | 0              |
| q <sub>2</sub> | 0              | 1              | 1              | 0              |
| q <sub>3</sub> | 0              | 1              | 0              | 1              |
| q <sub>4</sub> | 0              | 0              | 1              | 1              |

Vị trí 3:

$$TA = \{A_1, A_3, A_2\}, \quad TQ = \{q_2, q_1\},$$

$$BA = \{A_4\}, \quad BQ = \{\},$$

$$OQ = \{q_4, q_3\}$$

$$CTQ_3 = acc_1(q_1) + acc_2(q_1) + acc_3(q_1)$$

$$acc_1(q_2) + acc_2(q_2) + acc_3(q_2) = 50$$

$$CBQ_3 = 0$$

$$COQ_3 = acc_1(q_3) + acc_2(q_3) + acc_3(q_3) +$$

$$acc_1(q_4) + acc_2(q_4) + acc_3(q_4) = 78$$

$$Z = CTQ * CBQ - COQ^2 = -6084$$

Site1

$$acc_1(q_1)=15$$

$$acc_1(q_2)=5$$

$$acc_1(q_3)=25$$

$$acc_1(q_4)=3$$

Site2

$$acc_2(q_1)=20$$

$$acc_2(q_2)=0$$

$$acc_2(q_3)=25$$

$$acc_2(q_4)=0$$

Site3

$$acc_3(q_1)=10$$

$$acc_3(q_2)=0$$

$$acc_3(q_3)=25$$

$$acc_3(q_4)=0$$

## *vd Thuật toán phân mảnh dọc*

---

- ❑ Vị trí 1:  $Z = -2025$
- ❑ Vị trí 2:  $Z = 3311$
- ❑ Vị trí 3:  $Z = -6084$
- ❑ Như vậy vị trí 2 có chi phí là lớn nhất
- ❑ Quan hệ PROJ chia thành 2 mảnh:  
$$\text{PROJ}_1 \{A_1, A_3\} = \text{PROJ}_1 \{\underline{\text{PNO}}, \text{BUDGET}\}$$
$$\text{PROJ}_2 \{A_1, A_2, A_4\} = \text{PROJ}_2 \{\underline{\text{PNO}}, \text{PNAME}, \text{LOC}\}$$



## *vdThuật toán phân mảnh dọc*

---

### PROJ

| PNO | PNAME            | BUDGET | LOG      |
|-----|------------------|--------|----------|
| P1  | Instrumentation  | 150000 | Montreal |
| P2  | Database Develop | 135000 | NewYork  |
| P3  | CAD/CAM          | 250000 | NewYork  |
| P4  | Maintenance      | 310000 | Paris    |

### PROJ1

| PNO | BUDGET |
|-----|--------|
| P1  | 150000 |
| P2  | 135000 |
| P3  | 250000 |
| P4  | 310000 |

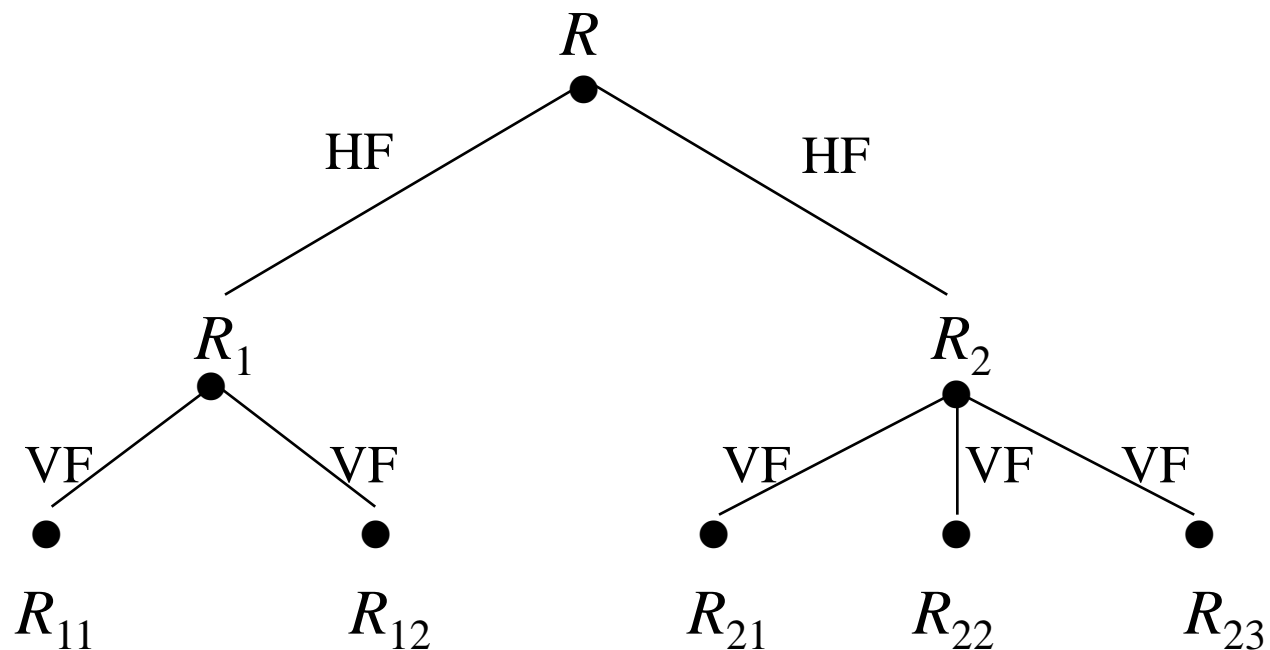
### PROJ2

| PNO | PNAME            | LOG      |
|-----|------------------|----------|
| P1  | Instrumentation  | Montreal |
| P2  | Database Develop | NewYork  |
| P3  | CAD/CAM          | NewYork  |
| P4  | Maintenance      | Paris    |

- Một quan hệ  $R_i$  được xác định trên tập thuộc tính  $A$  và khóa  $K_i$  tạo ra các mảnh dọc  $F_R = \{R_1, R_2, \dots, R_r\}$ .
- Tính đầy đủ
  - Được đảm bảo bởi giải thuật phân mảnh, gán mỗi thuộc tính của  $A$  cho mỗi mảnh
  - $A$  thỏa mãn:
    - $A = \bigcup A_{R_i}$
- Tính tái cấu trúc
  - Tái cấu trúc  $R = R_1 \bowtie \dots \bowtie R_n$  qua phép nối
- Tính tách rời
  - Các thuộc tính phải tách rời trong VF
  - Các ID của các bộ không được coi là chồng chéo vì chúng được duy trì bởi hệ thống
  - Các khóa không được coi là chồng chéo

# Phân mảnh lai (Hybrid Fragmentation)

- Sử dụng cả phân mảnh dọc và phân mảnh ngang=> Phân mảnh lai
- Có hai hướng: Phân mảnh ngang rồi đến phân mảnh dọc hoặc ngược lại
- Kiểm tra tính đúng đắn của Phân mảnh lai:  
Khôi phục phân đoạn lai
  - + VF; Dùng phép kết nối
  - + HF: Dùng phép hợp



# Nhân bản và cấp phát dữ liệu (1)

- **Nhân bản:** các mảnh nào sẽ được lưu trữ nhiều bản sao
  - Nhân bản toàn phần
  - Nhân bản có lựa chọn
- **Cấp phát:** Mảnh nào được lưu ở vị trí nào?

- Phát biểu bài toán
  - Cho
    - $F = \{F_1, F_2, \dots, F_n\}$  các mảnh
    - $S = \{S_1, S_2, \dots, S_m\}$  các vị trí mạng
    - $Q = \{q_1, q_2, \dots, q_q\}$  các ứng dụng
  - Tìm phân bổ tối ưu của  $F$  tới  $S$ .
- Tính tối ưu
  - Chi phí tối thiểu
    - Truyền thông + lưu trữ + xử lý (read & update)
    - Chi phí về thời gian
  - Hiệu năng
    - Thời gian đáp ứng và/hoặc thông lượng
  - Các ràng buộc
    - Các ràng buộc tại mỗi vị trí (lưu trữ & xử lý)

## Thông tin dữ liệu

- Chọn lựa các mảnh:  $Sel_i(F_j)$ : Số lượng các bộ của  $F_j$  đc truy xuất xử lý
- Kích cỡ các mảnh:  $Size(F_j) = Card(F_j) * length(F_j)$

## Thông tin ứng dụng

- $RR_{ij}$ : số các truy cập đọc từ ứng dụng  $q_i$  đến mảnh  $F_j$
- $UR_{ij}$ : số các truy cập cập nhật từ ứng dụng  $q_i$  đến mảnh  $F_j$
- $u_{ij}$  một phần tử ma trận chỉ ra truy vấn nào cập nhật mảnh nào

$$u_{ij} = \begin{cases} 1 & \text{if query } q_i \text{ updates fragment } F_j \\ 0 & \text{otherwise} \end{cases}$$

- $r_{ij}$  một phần tử

$$r_{ij} = \begin{cases} 1 & \text{if query } q_i \text{ retrieves from fragment } F_j \\ 0 & \text{otherwise} \end{cases}$$

## Thông tin vị trí

- $USC_k$  chi phí đơn vị cho lưu trữ dữ liệu tại vị trí  $S_k$
- $LPC_k$  chi phí cho xử lý một đơn vị dữ liệu tại vị trí  $S_k$

## Thông tin mạng

- Chi phí/khung truyền thông giữa hai vị trí  $S_i$  và  $S_j$ :  $g_{ij}$
- Kích cỡ khung:  $fsize()$

Dạng tổng quát

$\min(\text{Tổng chi phí})$

đối với

ràng buộc thời gian đáp ứng

ràng buộc lưu trữ

ràng buộc xử lý

▣ Các biến quyết định

$$x_{ij} = \begin{cases} 1 & \text{nếu đoạn } F_i \text{ được lưu tại vị trí } S_j \\ 0 & \text{trường hợp khác} \end{cases}$$



# Cấp phát các mảnh dữ liệu (1)

- Hàm chi phí tổng có hai thành phần: lưu trữ và xử lý

$$TOC = \sum_{S_k \in S} \sum_{F_j \in F} STC_{jk} + \sum_{q_i \in Q} QPC_i$$

- Chi phí lưu trữ của mảnh  $F_j$  tại vị trí  $S_k$

$$STC_{jk} = USC_k * size(F_j) * x_{ij}$$

ở đó  $USC_k$  là chi phí lưu trữ đơn vị tại vị trí  $S_k$

- Chi phí xử lý truy vấn của một truy vấn  $q_i$  bao gồm hai thành phần chi phí xử lý PC và chi phí truyền dẫn TC

$$QPC_i = PC_i + TC_i$$

- Chi phí xử lý là tổng của 3 thành phần
  - Chi phí truy cập (AC), chi phí đảm bảo toàn vẹn (IE), chi phí điều khiển tương tranh CC

$$PC_i = AC_i + IE_i + CC_i$$

- Chi phí truy cập

$$AC_i = \sum_{s_k \in S} \sum_{F_j \in F} (UR_{ij} + RR_{ij}) * X_{ij} * LPC_k$$

- $LPC_k$ : Chi phí xử lý đơn vị tại vị trí k

- Chi phí đảm bảo toàn vẹn và điều khiển tương tranh (tính toán tương tự dựa vào các ràng buộc cụ thể)

# Cấp phát các mảnh dữ liệu (3)

- Chi phí truyền dẫn bao gồm hai thành phần: chi phí xử lý cập nhật và chi phí xử lý truy vấn

$$TC_i = TCU_i + TCR_i$$

- Chi phí cập nhật

$$TCU_i = \sum_{S_k \in S} \sum_{F_j \in F} u_{ij} * (\text{update message cost} + \text{acknowledgment cost})$$

- Chi phí truy vấn

$$TCR_i = \sum_{F_j \in F} \min_{S_k \in S} (x_{jk} * (\text{cost retrieval request} + \text{cost sending back result}))$$

- Mô hình hóa các ràng buộc

- Ràng buộc thời gian đáp ứng cho truy vấn  $q_i$
- Thời gian thực thi của  $q_i \leq$  thời gian đáp ứng cực đại cho phép của  $q_i$
- Các ràng buộc về lưu trữ cho vị trí  $S_k$

$$\sum_{F_j \in F} \text{storage requirement of } F_j \text{ at } S_k \leq \text{storage capacity of } S_k$$

- Các ràng buộc về xử lý của vị trí  $S_k$

$$\sum_{q_i \in Q} \text{processing load of } q_i \text{ at site } S_k \leq \text{processing capacity of } S_k$$

# Cấp phát các mảnh dữ liệu (5)

- Độ phức tạp của bài toán cấp phát dữ liệu là NP-complete
  - Tương đồng với các bài toán trong lĩnh vực khác
  - Bài toán knapsack
  - Bài toán luồng mạng
  - Do đó, các giải pháp từ các lĩnh vực này có thể được sử dụng
  - Sử dụng các kinh nghiệm khác nhau để giảm không gian tìm kiếm
    - Giả sử rằng tất cả các phân chia ứng cử được xác định cùng nhau với các chi phí và các lời ích liên quan ở góc độ xử lý truy vấn
    - Vấn đề được rút gọn thành việc tìm phân chia tối ưu và vị trí cho mỗi quan hệ
    - Bỏ qua việc nhân bản ở bước đầu tiên và tìm giải pháp tối ưu cho bài toán trong trường hợp không nhân bản
    - Vấn đề nhân bản sau đó được xử lý ở bước thứ 2

# Tổng kết

- Vấn đề thiết kế
- Các chiến lược thiết kế (trên xuống, dưới lên)
- Phân mảnh dữ liệu (ngang, đứng)
- Cấp phát và nhân bản các phân mảnh

## ■ So sánh các lựa chọn nhân bản

|                      | Full-replication     | Partial-replication | Partitioning         |
|----------------------|----------------------|---------------------|----------------------|
| QUERY PROCESSING     | Easy                 | ← Same Difficulty → |                      |
| DIRECTORY MANAGEMENT | Easy or Non-existent | ← Same Difficulty → |                      |
| CONCURRENCY CONTROL  | Moderate             | Difficult           | Easy                 |
| RELIABILITY          | Very high            | High                | Low                  |
| REALITY              | Possible application | Realistic           | Possible application |

PROJ<sub>1</sub>

| PNO | PNAME               | BUDGET | LOC      |
|-----|---------------------|--------|----------|
| P1  | Instrumentatio<br>n | 150000 | Montreal |

PROJ<sub>2</sub>

| PNO | PNAME                | BUDGET | LOC      |
|-----|----------------------|--------|----------|
| P2  | Database<br>Develop. | 135000 | New York |

PROJ<sub>4</sub>

| PNO | PNAME   | BUDGET | LOC      |
|-----|---------|--------|----------|
| P3  | CAD/CAM | 250000 | New York |

PROJ<sub>6</sub>

| PNO | PNAME           | BUDGET | LOC   |
|-----|-----------------|--------|-------|
| P4  | Maintenanc<br>e | 310000 | Paris |