

Chương II Các hệ cơ sở dữ liệu phân tán

Nguyễn Kim Anh
anhnk-fit@mail.hut.edu.vn

Bộ môn Hệ thống Thông tin, SoICT

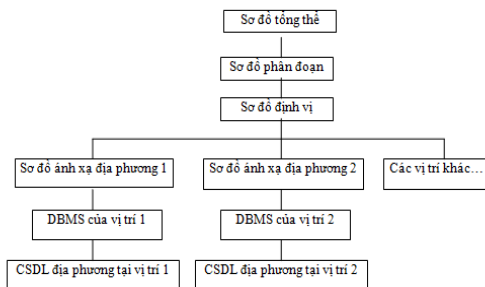
1

Nội dung

- Tổng quan về các hệ CSDLPT
- Phân đoạn dữ liệu
- Biểu diễn các yêu cầu với các mức trong suốt khác nhau
- **Thiết kế CSDLPT**
- Tối ưu hóa CSDL phân tán
- Quản trị giao dịch và điều khiển tương tranh

2

Kiến trúc của CSDL phân tán



3

Các cách tiếp cận đ/v bài toán thiết kế CSDLPT

- Trên-xuống:
 - Thiết kế tổng thể từ đầu
 - Áp dụng đối với hệ CSDLPT thuần nhất
- Dưới-lên
 - Các CSDL địa phương đã tồn tại và đang được truy cập và khai thác
 - Áp dụng đối với hệ CSDLPT không thuần nhất

Phân tích bài toán thiết kế CSDLPT

- Thiết kế CSDL phân tán
 - Thiết kế phân đoạn dữ liệu
 - Thiết kế sắp chỗ dữ liệu
- Các thông tin cần sử dụng cho bài toán thiết kế phân tán có thể chia thành bốn loại:
 - Thông tin về CSDL
 - Thông tin về ứng dụng
 - Thông tin về mạng
 - Thông tin về hệ thống máy tính

Thiết kế phân đoạn dữ liệu

1 - Phân đoạn dữ liệu là gì?

Việc chia/tách các quan hệ tổng thể thành nhiều quan hệ nhỏ hơn hay nhiều đoạn được gọi là phân đoạn dữ liệu.

6

Thiết kế phân đoạn dữ liệu

2 – Các lý do phải phân đoạn dữ liệu:

- Khung nhìn hoặc đơn vị truy xuất của các ứng dụng không phải là toàn bộ quan hệ tổng thể mà thường là một đoạn.
- Việc phân tách một quan hệ thành nhiều đoạn, mỗi đoạn được xử lý như một đơn vị, sẽ cho phép thực hiện nhiều giao dịch đồng thời.
- Việc phân đoạn các quan hệ sẽ cho phép thực hiện song song một câu truy vấn bằng cách tách nó ra thành một tập các câu truy vấn con thực hiện trên các đoạn.

7

Thiết kế phân đoạn dữ liệu

3 – Các kiểu phân đoạn:

- Phân đoạn ngang (horizontal fragmentation)
 - **phân đoạn ngang đơn giản** (primary horizontal fragmentation)
 - **phân đoạn ngang suy diễn** (derived horizontal fragmentation)
 - Phân đoạn dọc (vertical fragmentation).
 - Phân đoạn hỗn hợp (híbrid fragmentation)
- *Chú ý: Việc phân đoạn dữ liệu phải được gắn liền với việc định vị dữ liệu*

8

Thiết kế phân đoạn dữ liệu

4- Các điều kiện đối với một phân đoạn

- Việc phân đoạn một quan hệ tổng thể phải tuân theo một số điều kiện nhất định để khi tái thiết lại quan hệ cũ vẫn bảo đảm ngữ nghĩa của nó.
- Một phân đoạn đúng đắn phải thỏa mãn ba điều kiện sau:
 - *Tính đầy đủ (completeness)*
 - *Tính tái thiết được (reconstruction)*
 - *Tính tách biệt (disjointness):*

Thiết kế phân đoạn dữ liệu

• Thiết kế phân đoạn dữ liệu gồm:

- Thiết kế phân đoạn ngang đơn giản
- Thiết kế phân đoạn ngang suy diễn
- Thiết kế phân đoạn dọc
- Thiết kế phân đoạn hỗn hợp

Thiết kế phân đoạn ngang đơn giản

- Một đoạn ngang được xác định bằng phép chọn trên quan hệ tổng thể .

$$R_i = \sigma_{F_i}(R); 1 \leq i \leq n$$
 - F_i là điều kiện chọn/tiêu chuẩn của đoạn R_i
- Phân đoạn ngang đơn giản là đúng đắn nếu thỏa mãn ba điều kiện sau:
 - *Tính đầy đủ* : $\forall b \in R, \exists 1 \leq i \leq n, b \text{ thỏa } F_i$
 - *Tính tái thiết được* : $R = R_1 \cup R_2 \cup \dots \cup R_n$
 - *Tính tách biệt* : F_i và F_j là loại trừ nhau với $i \neq j$

Thiết kế phân đoạn ngang đơn giản

- Do điều kiện chọn của phép chọn là một biểu thức logic trên các phép so sánh đơn giản, từ đó:
 - phân đoạn ngang đơn giản của một quan hệ được thực hiện dựa trên các vị từ được định nghĩa trên các thuộc tính của quan hệ.
 - Cần xác định tập các vị từ đơn giản và ghép các vị từ đơn giản để hình thành các tiêu chuẩn của các đoạn ngang.

Thiết kế phân đoạn ngang đơn giản

*Thông tin về ứng dụng

Các vị từ đơn giản (phép so sánh giữa 1 thuộc tính và 1 giá trị hằng) được sử dụng trong các truy vấn.

- Chỉ phân tích các ứng dụng quan trọng để xác định các vị từ này.

Giả sử phân đoạn ngang đơn giản quan hệ R (A_1, A_2, \dots, A_n), với A_i là thuộc tính được xác định trên miền D_i .

13

Thiết kế phân đoạn ngang đơn giản

- **Vị từ đơn giản (simple predicate)** p_i được định nghĩa trên R có dạng: $A_i \theta \text{ value}$

– θ là một trong các phép so sánh $=, \neq, <, \leq, >, \geq$ và value được chọn từ miền trị của A_i ($\text{value} \in D_i$)

– Ký hiệu P_r là tập các vị từ đơn giản được định nghĩa trên quan hệ R . Các phần tử của P_r được ký hiệu là p_i .

14

Thiết kế phân đoạn ngang đơn giản

- **Vị từ hội tối thiểu (minterm predicate)** m_j đối với tập các vị từ đơn giản $P_r = \{p_1, p_2, \dots, p_m\}$ là phép hội của tất cả các vị từ xuất hiện trong P_r (ở dạng tự nhiên hoặc ở dạng phủ định):

$$m_j = \bigwedge_i p_i^*, 1 \leq i \leq m$$

với $p_i^* = p_i$ hoặc $p_i^* = \neg p_i$ và $m_j \neq \text{false}$

Gọi tập các vị từ hội tối thiểu là: $M = \{m_1, m_2, \dots, m_z\}$

- Mỗi một đoạn ngang của R có tiêu chuẩn là một vị từ hội tối thiểu $\rightarrow R$ được tách thành z đoạn tương ứng với $M = \{m_1, m_2, \dots, m_z\}$

15

Thiết kế phân đoạn ngang đơn giản

* Các bước thiết kế phân đoạn ngang đơn giản:

– **Bước 1:** Tìm tập các vị từ đơn giản P_r . (P_r là đầy đủ và tối thiểu).

– **Bước 2:** Xác định tập các vị từ hội tối thiểu có thể được định nghĩa trên các vị từ của P_r .

16

Thiết kế phân đoạn ngang đơn giản

- Một vị từ đơn giản p_i được gọi là *thích đáng* (*relevant*) đối với một tập P_r các vị từ đơn giản, nếu tồn tại ít nhất hai vị từ hội tối thiểu m_k và m_j của P_r mà các biểu thức của chúng chỉ khác nhau ở p_i (tức là m_k chứa p_i và m_j chứa $\neg p_i$) và tồn tại ít nhất một ứng dụng tham chiếu một cách khác nhau đến hai đoạn f_k và f_j (tương ứng với m_k và m_j).

17

Thiết kế phân đoạn ngang đơn giản

- Một tập các vị từ đơn giản P_r được gọi là *đầy đủ* (*complete*) nếu và chỉ nếu bất kỳ hai bộ nào thuộc cùng một đoạn được định nghĩa theo P_r đều được tham chiếu với cùng một xác suất bởi mọi ứng dụng.
- Một tập các vị từ đơn giản P_r được gọi là *tối thiểu* (*minimal*) nếu tất cả các vị từ của nó là các vị từ thích đáng.
- Cho $P_r = \{p_1, p_2, \dots, p_m\}$ là một tập các vị từ đơn giản. Khi đó P_r biểu diễn phân đoạn đúng đắn và hiệu quả nếu P_r là *đầy đủ* và *tối thiểu*.
(Chứng minh????)

18

Ví dụ:

- Ví dụ:** Xét cơ sở dữ liệu của một công ty máy tính được tổ chức như sau:
 - NHANVIEN (MANV, TENNV, CHUCVU):** quan hệ này chứa dữ liệu về nhân viên của công ty.
 - TLUONG (CHUCVU, LUONG):** quan hệ này chứa dữ liệu liên quan về lương và chức vụ của nhân viên.
 - DUAN (MADA, TENDA, NGANSACH):** quan hệ này chứa dữ liệu về các dự án mà công ty đang phát triển.
 - HOSO (MANV, MADA, NHIEMVU, THOIGIAN):** quan hệ này chứa dữ liệu về hồ sơ của nhân viên được phân công thực hiện dự án.

19

Ví dụ:

NHANVIEN (E)

MANV	TENNV	CHUCVU
A1	Nam	Phân tích HT
A2	Trung	Lập trình viên
A3	Đông	Phân tích HT
A4	Bắc	Phân tích HT
A5	Tây	Lập trình viên
A6	Hùng	Kỹ sư điện
A7	Dũng	Phân tích HT
A8	Chiến	Thiết kế DL

20

HOSO (G)

MA NV	MA DA	NHIEMVU	THOI GIAN
A1	D1	Quản lý	12
A2	D1	Phân tích	34
A2	D2	Phân tích	6
A3	D3	Kỹ thuật	12
A3	D4	Lập trình	10
A4	D2	Quản lý	6
A5	D2	Quản lý	20
A6	D4	Kỹ thuật	36
A7	D3	Quản lý	48
A8	D3	Lập trình	15

Ví dụ:

MADA	TENDA	NGANSACH
D1	CSDL	20000
D2	CẢI ĐẶT	12000
D3	BẢO TRÌ	28000
D4	PHÁT TRIỂN	25000

DUAN (J)

CHUCVU	LUONG
Kỹ sư điện	1000
Phân tích HT	2500
Lập trình viên	3000
Thiết kế DL	4000

TIENLUONG (S)

21

Ví dụ:

Thực hiện phân đoạn ngang đơn giản quan hệ DUAN:

***Thông tin về ứng dụng**

➢ Chỉ phân tích các ứng dụng quan trọng truy cập đến quan hệ DUAN để xác định các vị từ đơn giản:

select * from DUAN where NGANSACH > 20000

- Các vị từ đơn giản được sử dụng trong các truy vấn: p_I : NGANSACH > 20000

- $P_I = \{p_I\}$ là đầy đủ và tối thiểu

- Tập các vị từ hội tối thiểu là: $M = \{m_1, m_2\}$ với

$$m_1 = p_I = \text{NGANSACH} > 20000$$

$$m_2 = \neg p_I = \text{NGANSACH} \leq 20000$$

Ví dụ:

Ví dụ về phân mảnh ngang: xét các phép toán đại số quan hệ sau:

DUAN 1 = $\sigma_{\text{NGANSACH} \leq 20000}$ (DUAN)

DUAN 2 = $\sigma_{\text{NGANSACH} > 20000}$ (DUAN)

DUAN 1

MADA	TENDA	NGANSACH
D1	CSDL	20000
D2	CẢI ĐẶT	12000

DUAN 2

MADA	TENDA	NGANSACH
D3	BẢO TRÌ	28000
D4	PHÁT TRIỂN	25000

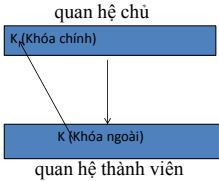
Dễ thấy, các mảnh thỏa mãn tính tái thiết được và tính đầy đủ

DUAN 1 \subseteq DUAN ; DUAN 2 \subseteq DUAN ; DUAN = DUAN 1 \cup DUAN 2

23

Thiết kế phân đoạn ngang suy diễn

- Phân đoạn ngang suy diễn đối với một quan hệ tổng thể (quan hệ thành viên) là một phân đoạn ngang được suy diễn ra từ sự phân đoạn ngang của một quan hệ tổng thể khác (quan hệ chủ)

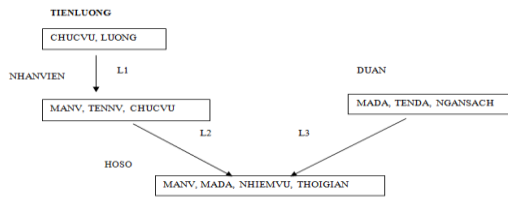


24

Thiết kế phân đoạn ngang suy diễn

* Thông tin về cơ sở dữ liệu:

- Trong sơ đồ quan niệm tổng thể, các quan hệ được móc nối với nhau thông qua các liên kết.



25

Thiết kế phân đoạn ngang suy diễn

* Thông tin về ứng dụng

Các vị từ nối (phép so sánh giữa 1 thuộc tính của quan hệ này với 1 thuộc tính của quan hệ khác) được sử dụng trong các truy vấn.

- Chỉ phân tích các ứng dụng quan trọng để xác định các vị từ này.
- 1 vị từ nối ứng với 1 phép nối 2 quan hệ dựa trên mỗi quan hệ khóa chính-khóa ngoài.
- Phân tích phép nối 2 quan hệ được phân đoạn ngang:

$$R \mid >< \mid S = \bigcup_{i,j} (R_i \mid >< \mid S_j)$$

Thiết kế phân đoạn ngang suy diễn

- Một đoạn ngang được xác định bằng phép nửa-kết nối/nối-nửa của quan hệ thành viên (R) và đoạn tương ứng của quan hệ chủ (S).

$$R_i = R \mid >< \mid S_i; 1 \leq i \leq n$$
 - điều kiện nối F_i có dạng: $KC_S = KN_R$, trong đó KC_S là khóa chính của S và KN_R là khóa ngoài trong R
- Phân đoạn ngang suy diễn là đúng đắn nếu thỏa mãn ba điều kiện sau:
 - Tính đầy đủ: R chứa khóa ngoài tham chiếu đến khóa chính của S (ràng buộc toàn vẹn tham chiếu)
 - Tính tái thiết được: $R = R_1 \cup R_2 \cup \dots \cup R_n$
 - Tính tách biệt: KC_S là khóa chính của S

27

Ví dụ :

- Ví dụ về phân đoạn ngang suy diễn:
 - $DUAN(MADA, TENDA, NGANSACH)$: quan hệ chủ chứa khóa chính MADA
 - $HOSO(MANV, MADA, NHIEMVU, THOIGIAN)$: quan hệ thành viên chứa khóa ngoài MADA tham chiếu đến DUAN
 $\Rightarrow HOSO_i = HOSO \mid >< \mid DUAN_i; 1 \leq i \leq 2$
 - $NHANVIEN(MANV, TENNV, CHUCVU)$: quan hệ chủ chứa khóa chính MANV
 - $HOSO(MANV, MADA, NHIEMVU, THOIGIAN)$: quan hệ thành viên chứa khóa ngoài MANV tham chiếu đến NHANVIEN
 $\Rightarrow HOSO_i = HOSO \mid >< \mid NHANVIEN_i; 1 \leq i \leq 2$
- Quyết định phân đoạn HOSO ?????

28

Thiết kế phân đoạn đọc

- Một đoạn đọc được xác định bằng phép chiếu trên quan hệ tổng thể.

$$R_i = \Pi_{U_i}(R); 1 \leq i \leq n$$

- U_i là tập thuộc tính của đoạn R_i
- Phân đoạn đọc là đúng đắn nếu thỏa mãn ba điều kiện sau:
 - Tính đầy đủ: $U = U_1 \cup U_2 \cup \dots \cup U_n$
 - Tính tái thiết được: $R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_n$
 - Tính tách biệt: $U_i \cap U_j$ là tối thiểu với $i \neq j$

29

Thiết kế phân đoạn đọc

- Nhóm thuộc tính:
 - gán mỗi thuộc tính cho một đoạn
 - nối/ghép 2 đoạn nhỏ lại cho đến khi thỏa một tiêu chuẩn nào đó
- Tách đoạn: Bắt đầu với
 - gán quan hệ tổng thể là đoạn
 - tách đoạn thành 2 đoạn nhỏ hơn dựa trên các thao tác truy xuất của các ứng dụng trên các thuộc tính
- Do phân đoạn đọc đặt vào một đoạn các thuộc tính thường được truy xuất cùng nhau → cần một đại lượng đo mức độ liên quan hay độ gần gũi dựa trên khái niệm “truy xuất cùng nhau”.

Thiết kế phân đoạn đọc

*Thông tin về ứng dụng

Các thuộc tính của quan hệ xuất hiện trong các truy vấn và tần suất của truy vấn.

- Giả sử $Q = \{q_1, q_2, \dots, q_n\}$ là tập các truy vấn (các ứng dụng) thực hiện trên quan hệ $R(A_1, A_2, \dots, A_n)$.
- Với mỗi q_i và mỗi thuộc tính A_j , giá trị sử dụng thuộc tính $use(q_i, A_j)$ được định nghĩa như sau:

$$use(q_i, A_j) = \begin{cases} 1 & \text{nếu } A_j \text{ xuất hiện trong } q_i \\ 0 & \text{ngược lại} \end{cases}$$
- Các vectơ $use(q_i, \bullet)$ cho mỗi ứng dụng q_i được xác định nếu nhà thiết kế biết được các ứng dụng sẽ chạy trên CSDL.

Thiết kế phân đoạn đọc

- Độ gần gũi giữa 2 thuộc tính A_i, A_j được định nghĩa là:

$$aff(A_i, A_j) = \sum_{use(q_k, A_i)=1 \wedge use(q_k, A_j)=1} ref_i(q_k) acc_j(q_k)$$
 - q_k truy xuất đến cả 2 thuộc tính (A_i, A_j) ,
 - $ref_i(q_k)$ là số lần truy xuất đến các thuộc tính (A_i, A_j) với mỗi lần thực hiện ứng dụng q_k tại trạm l và
 - $acc_j(q_k)$ là tần suất của ứng dụng q_k được phát ra tại trạm l .

Ví dụ:

$$\begin{array}{c}
 q_1 \\
 q_2 \\
 q_3 \\
 q_4
 \end{array}
 \begin{pmatrix}
 & A_1 & A_2 & A_3 & A_4 \\
 & 1 & 0 & 1 & 0 \\
 & 0 & 1 & 1 & 0 \\
 & 0 & 1 & 0 & 1 \\
 & 0 & 0 & 1 & 1
 \end{pmatrix}$$

Ví dụ:

- Để đơn giản chúng ta giả sử rằng $\text{ref}_l(q_k) = 1$ đối với tất cả q_k và l và tần suất ứng dụng là:

$\text{acc}_1(q_1) = 15$	$\text{acc}_2(q_1) = 20$	$\text{acc}_3(q_1) = 10$
$\text{acc}_1(q_2) = 5$	$\text{acc}_2(q_2) = 0$	$\text{acc}_3(q_2) = 0$
$\text{acc}_1(q_3) = 25$	$\text{acc}_2(q_3) = 25$	$\text{acc}_3(q_3) = 25$
$\text{acc}_1(q_4) = 3$	$\text{acc}_2(q_4) = 0$	$\text{acc}_3(q_4) = 0$
- Do chỉ có 1 ứng dụng q_1 truy xuất đến cả 2 thuộc tính (A_1, A_3), độ gần gũi giữa hai thuộc tính A_1 và A_3 là:

$$\text{aff}(A_1, A_3) = \sum_{k=1}^1 \sum_{l=1}^3 \text{acc}_l(q_k) = \text{acc}_1(q_1) + \text{acc}_2(q_1) + \text{acc}_3(q_1) = 45$$
- Tương tự tính cho các cặp thuộc tính khác, ta có ma trận gần gũi sau:

Ví dụ:

$$\begin{array}{c}
 A_1 \\
 A_2 \\
 A_3 \\
 A_4
 \end{array}
 \begin{pmatrix}
 & A_1 & A_2 & A_3 & A_4 \\
 & 45 & 0 & 45 & 0 \\
 & 0 & 80 & 5 & 75 \\
 & 45 & 5 & 53 & 3 \\
 & 0 & 75 & 3 & 78
 \end{pmatrix}$$

Ví dụ:

- Ví dụ về phân đoạn dọc quan hệ [HOSO\(MANV, MADA, NHIEMVU, THOIGIAN\)](#):

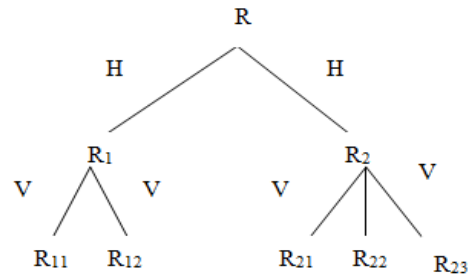
$$\text{HOSO}_1 = \Pi_{\text{MANV}, \text{NHIEMVU}, \text{TID}}(\text{HOSO})$$

$$\text{HOSO}_2 = \Pi_{\text{MADA}, \text{THOIGIAN}, \text{TID}}(\text{HOSO})$$

Thiết kế phân đoạn hỗn hợp

- Là phân đoạn được thực hiện bằng cách áp dụng một cách đệ quy/đồng thời cả phân đoạn ngang và phân đoạn dọc

Thiết kế phân đoạn hỗn hợp



37

Ví dụ:

- Ví dụ về phân đoạn hỗn hợp quan hệ
 $HOSO(MANV, MADA, NHIEMVU, THOIGIAN)$:
 $HOSO_1 = \Pi_{MANV, NHIEMVU, TID}(HOSO) \mid > < NHANVIEN_1$
 $HOSO_2 = \Pi_{MANV, NHIEMVU, TID}(HOSO) \mid > < NHANVIEN_2$
 $HOSO_3 = \Pi_{MADA, THOIGIAN, TID}(HOSO) \mid > < DUAN_1$
 $HOSO_4 = \Pi_{MADA, THOIGIAN, TID}(HOSO) \mid > < DUAN_2$

39

Thiết kế sắp chỗ dữ liệu

- Giả sử đã có một tập các đoạn $F = \{F_1, F_2, \dots, F_n\}$ và một mạng bao gồm các trạm $S = \{S_1, S_2, \dots, S_m\}$ trên đó có một tập các ứng dụng $Q = \{q_1, q_2, \dots, q_q\}$ đang thực hiện.
- Bài toán sắp chỗ với mục đích tìm một cách sắp chỗ "tối ưu" F trên S .
- Qui ước:
 - chỉ số đoạn: i
 - chỉ số trạm: j
 - chỉ số truy vấn/ứng dụng: k

Thiết kế sắp chỗ dữ liệu

- Tính tối ưu có thể được xác định dựa trên hai độ đo:
 - **Tổng chi phí** : tìm một lược đồ sắp chỗ với tổng chi phí nhỏ nhất.
 - **Hiệu năng**: duy trì một hiệu năng cao
- Bài toán sắp chỗ tổng quát là một bài toán NP-đầy đủ (NP-complete)
- cần tìm các thuật giải heuristic tốt để có lời giải gần tối ưu.

Yêu cầu về thông tin

- **Thông tin về CSDL**
 - Độ chọn lọc của một đoạn F_i ứng với câu truy vấn q_k : $sel_k(F_i)$ - số lượng các bộ của F_i cần được truy xuất để xử lý q_k
 - Kích thước của một đoạn F_i được cho bởi
 - $size(F_i) = card(F_i) * length(F_i)$
 - $length(F_i)$: chiều dài (tính theo byte) của một bộ trong đoạn F_i .
 - $card(F_i)$: số bộ trong đoạn F_i .

Yêu cầu về thông tin

- **Thông tin về ứng dụng**
 - Số truy xuất đọc/cập nhật đoạn F_i đối với câu truy vấn q_k : RR_{ik}/UR_{ik} .
 - Trạng gốc của câu truy vấn q_k : $g(q_k)$

Yêu cầu về thông tin

- **Thông tin về vị trí/trạm**
 - khả năng lưu trữ và
 - khả năng xử lý
 - + Chi phí để lưu trữ 1 đơn vị dữ liệu tại vị trí S_j : USC_j .
 - + Chi phí xử lý 1 đơn vị công việc tại vị trí S_j : LPC_j
- **Thông tin về mạng**
 - chi phí truyền 1 đơn vị dữ liệu giữa hai vị trí S_j và S_l : g_{jl}

Mô hình sắp chỗ

- Min (Total Cost)
với ràng buộc thời gian đáp ứng,
ràng buộc lưu trữ và
ràng buộc xử lý.
- Biến quyết định x_{ij} được định nghĩa là

$$x_{ij} = \begin{cases} 1 & \text{nếu } F_i \text{ được đặt tại vị trí } S_j \\ 0 & \text{ngược lại} \end{cases}$$

Mô hình sắp chỗ

- Hàm tổng chi phí có hai thành phần:
 - phần xử lý truy vấn tin và
 - phần lưu trữ
$$TOC = \sum_k QPC_k + \sum_{ij} STC_{ij}$$
- QPC_k : chi phí xử lý câu truy vấn q_k và
- STC_{ij} : chi phí lưu trữ đoạn F_i tại vị trí S_j

Mô hình sắp chỗ

- $STC_{ij} = USC_j * size(F_i) * x_{ij}$
- $QPC_k = PC_k + TC_k$
 - PC_k : chi phí xử lý đối với q_k
 - TC_k : chi phí truyền đối với q_k

Mô hình sắp chỗ

- Chi phí xử lý PC gồm chi phí truy xuất AC, chi phí đảm bảo toàn vẹn IE và chi phí điều khiển đồng thời CC:

$$PC_k = AC_k + IE_k + CC_k$$

$$AC_k = \sum_{ij} (UR_{ik} + RR_{ik}) * x_{ij} * LPC_j$$
 - $(UR_{ik} + RR_{ik})$: tổng số các truy xuất đọc và cập nhật đoạn F_i đối với q_k
 - Tính IE_k , CC_k và TC_k tương tự