

## VỀ MỘT THUẬT TOÁN CÀI ĐẶT CẢI BIẾN NHÓM PHÉP TOÁN CHIỀU - CHỌN - NỐI

NGUYỄN THANH THỦY

**T**RONG [1] một vài thuật toán cài đặt hữu hiệu phép toán mở rộng chiều-chọn-nối và đánh giá độ phức tạp của chúng đã được đưa ra. Trong bài này sẽ trình bày một vài cải biên của chúng kèm theo các đánh giá cụ thể độ phức tạp thời gian và bộ nhớ. Các khái niệm mở đầu có thể xem trong [1, 4].

### Thuật toán T.L.I.

Vào: Quan hệ R có tập thuộc tính W;  $A \subseteq W$ .

Quan hệ S có tập thuộc tính V;  $B \subseteq V$ ;  $C \subseteq W \cup V$ .

E, F tương ứng là biểu thức logic trên R, S.

Ra: RESULT =  $(\sigma_R(R) [A = B] \sigma_F(S)) [C]$

### Phương pháp

|\* qui ước: có chỉ dẫn lên các thuộc tính A, B tương ứng của R, S<sup>ab</sup> |.  
begin xác định  $P = R [A] \cap S [B]$

|\* có thể lấy  $P = R [A]$  hoặc  $P = S [B]$  \*|

for each  $e \in P$  do

begin truy nhập tới R để xác định  $M = \{r \in R \mid r [A] = e\}$  nhờ chỉ dẫn;

if not eof (M) then

begin for each  $r \in M$  do if r thỏa E then

add  $r[C \cap W]$  tới M1;

if not eof (M1) then

begin truy nhập tới S để xác định

$N = \{s \in S \mid s [B] = e\}$  nhờ chỉ dẫn;

if not eof(N) then

for each  $s \in N$  do

if s thỏa F then

begin  $s1 := s [C \cap V]$ ;

for each  $r1 \in M1$  do

begin  $t := (r1, s1)$

add t to RESULT

end

end

end

end

end

end

**Mệnh đề 1.** Độ phức tạp thời gian của thuật toán là:

$$\begin{aligned}
 & HS \cdot \sum_j \text{card}(R) \cdot s(A = e_j) + \sum_j \text{card}(R) \cdot s(A = e_j) \cdot \tau(E) + HS \cdot s(E) \cdot \sum_j \text{card}(S) \cdot s(B = e_j) + \\
 & + s(E) \cdot \sum_j \text{card}(S) \cdot s(B = e_j) \cdot \tau(F) + (HS + \eta) \text{card}(H) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A = B) + \\
 & + \gamma \left( \sum_j \text{card}(S) \cdot s(F) \cdot s(B = e_j) + \sum_j \text{card}(R) \cdot s(E) \cdot s(A = e_j) \right).
 \end{aligned}$$

Thuật toán đòi hỏi

$$k_1 \cdot \text{card}(R) + k_2 \cdot \text{card}(S) + (k_1 + k_2) \cdot \text{card}(R) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A=B) \text{ đơn vị nhớ ngoài.}$$

$$\max_j k_1 \cdot \text{card}(R) \cdot s(A = e_j) + \max_j k_1 \cdot \text{card}(R) \cdot s(A = e_j) \cdot s(E) +$$

$$+ s(E) \cdot \max_j k_2 \cdot \text{card}(S) \cdot s(B = e_j) \text{ đơn vị nhớ trong.}$$

Ở đây  $k_1, k_2$  lần lượt là độ dài mỗi bản ghi của quan hệ R, S, HS là tham số chỉ sự cân đối giữa các xử lý trong thiết bị trung tâm so với các xử lý vào ra hay so với các trao đổi dữ liệu giữa bộ nhớ trong và bộ nhớ ngoài.

$\eta$  là chi phí kết nối hai bộ  $r_1, s_1$ .

$\gamma$  là giá phải trả để chiếu lấy một số thuộc tính trong một bộ.

$\tau(E), \tau(F)$  tương ứng là thời gian trung bình để xác định một bộ  $r \in R$  cho trước thỏa E hay một bộ  $s \in S$  cho trước thỏa F. [2].

Chứng minh: Truy nhập tới R để xác định M đòi hỏi

$$HS \cdot \sum_j \text{card}(R) \cdot s(A = e_j) \text{ đơn vị thời gian}$$

$$\text{và } \max_j k_1 \cdot \text{card}(R) \cdot s(A = e_j) \text{ đơn vị nhớ trong.}$$

Kiểm tra các bộ trong M thỏa E chi phí

$$\sum_j \text{card}(R) \cdot s(A = e_j) \cdot \tau(E) \text{ đơn vị thời gian.}$$

$$\max_j k_1 \cdot \text{card}(R) \cdot s(A = e_j) \cdot s(E) \text{ đơn vị nhớ trong.}$$

Để truy nhập tới S, xác định N phải mất

$$s(E) \cdot HS \cdot \sum_j \text{card}(S) \cdot s(B = e_j) \text{ đơn vị thời gian,}$$

$$s(E) \cdot \max_j k_2 \cdot \text{card}(S) \cdot s(B = e_j) \text{ đơn vị nhớ trong.}$$

Thời gian kiểm tra các bộ  $s \in N$  thỏa F là

$$s(E) \cdot \sum_j \text{card}(S) \cdot s(B = e_j) \cdot \tau(F).$$

Kết nối các bộ  $r_1, s_1$  và cho vào RESULT mất

$$(HS + \eta) \text{card}(R) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A=B) \text{ đơn vị thời gian,}$$

$$(k_1 + k_2) \text{card}(R) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A=B) \text{ đơn vị nhớ ngoài.}$$

Chi phí để chiếu các bộ là

$$\gamma \left( \sum_j \text{card}(R) \cdot s(E) \cdot s(A = e_j) + \sum_j \text{card}(S) \cdot s(F) \cdot s(B = e_j) \right).$$

Tổng cộng chi phí bộ nhớ trong là

$$\max_j k_1 \cdot \text{card}(R) \cdot s(A = e_j) + \max_j k_1 \cdot \text{card}(R) \cdot s(A = e_j) \cdot s(E) + s(E) \cdot \max_j k_2 \cdot \text{card}(S) \cdot s(B = e_j).$$

Chi phí bộ nhớ ngoài là

$$k_1 \cdot \text{card}(R) + k_2 \cdot \text{card}(S) + (k_1 + k_2) \cdot \text{card}(R) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A=B).$$

Chi phí thời gian là

$$HS \cdot \sum_j \text{card}(R) \cdot s(A = e_j) + \sum_j \text{card}(R) \cdot s(A = e_j) \cdot \tau(E) + s(E) \cdot HS \cdot \sum_j \text{card}(S) \cdot s(B = e_j) +$$

$$+ \sum_j s(E) \cdot \text{card}(S) \cdot s(B = e_j) \cdot \tau(F) + (HS + \eta) \cdot \text{card}(R) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A=B)$$

$$+ \gamma \left( \sum_j \text{card}(R) \cdot s(E) \cdot s(A = e_j) + \sum_j \text{card}(S) \cdot s(F) \cdot s(B = e_j) \right). \text{ đ. p. c. m.}$$

Nhận xét 1: Đặt

$$\alpha_1 = \sum_j s(A = e_j), \quad \beta_1 = \sum_j s(B = e_j),$$

$$\gamma_1 = \max_j s(A = e_j), \quad \delta_1 = \max_j s(B = e_j),$$

$$\text{hằng} = \text{card}(R) \cdot \text{card}(S) \cdot s(E) \cdot s(F) \cdot s(A=B),$$

$$\text{constant} = \text{hằng} \cdot (k_1 + k_2) + k_1 \cdot \text{card}(R) + k_2 \cdot \text{card}(S),$$

$L = (HS + \eta) \cdot \text{hãng} + \gamma (\text{card}(R) \cdot s(E) \cdot \alpha_1 + \text{card}(S) \cdot s(F) \cdot \beta_1)$ .  
 Độ phức tạp thời gian (TC), độ phức tạp bộ nhớ (MC) của thuật toán T.L.1 là  
 $TC1 = \alpha_1 \cdot \text{card}(R) (HS + \tau(E)) + s(E) \cdot \beta_1 \cdot \text{card}(S) (HS + \tau(F)) + L$   
 $MC1 = \text{constant} + k_1 \cdot \gamma_1 \cdot \text{card}(R) (1 + s(E)) + s(E) \cdot k_2 \cdot \text{card}(S) \delta_1$   
 Độ phức tạp của thuật toán T.L.1 [1] là

$$(1) \begin{cases} TC2 = TC1 \\ MC2 = \text{constant} + k_1 \cdot \gamma_1 \cdot \text{card}(R) + s(E) \cdot k_2 \cdot \text{card}(S) \cdot \delta_1 (1 + S(F)). \end{cases}$$

Nếu thay đổi vai trò R, S (tương ứng E và F, A và B,  $k_1$  và  $k_2, \dots$ ) trong thuật toán T.L.1 [1], T.L.1 thì ta sẽ nhận được các thuật toán cải biên tương ứng.

với độ phức tạp TC3, MC3  
 TC4, MC4.

$$(2) TC3 = TC4 = \beta_1 \cdot \text{card}(S) (HS + \tau(F)) + s(F) \cdot \alpha_1 \cdot \text{card}(R) \cdot (HS + \tau(E) + L,$$

$$(3) MC4 = \text{constant} + k_2 \cdot \delta_1 \cdot \text{card}(S) (1 + s(F)) + s(F) \cdot k_1 \cdot \text{card}(R) \cdot \gamma_1,$$

$$(4) MC3 = \text{constant} + k_2 \cdot \delta_1 \cdot \text{card}(S) + s(F) \cdot k_1 \cdot \text{card}(R) \cdot \gamma_1 (1 + s(E)),$$

$$(5) TC1 \geq TC3 \Leftrightarrow \frac{\alpha_1 \cdot \text{card}(R) (HS + \tau(E))}{(1 - s(E))} > \frac{\beta_1 \cdot \text{card}(S) \cdot (HS + \tau(F))}{(1 - s(F))}$$

**Nhận xét 2:**

Hoàn toàn tương tự, nếu trong các thuật toán T.L.2, T.L.3 [1] thay đổi vai trò R và S thì lại nhận được những thuật toán cải biên của chúng. Tùy theo yêu cầu tiết kiệm thời gian hay bộ nhớ mà lựa chọn thuật toán thích hợp.

**Nhận xét 3:**

Trong thuật toán T.L.1, T.L.3 [1], T.L.1 ta đòi hỏi các tệp phải có các chỉ dẫn lên các thuộc tính chọn và các chỉ dẫn lên các thuộc tính nối. Khi yêu cầu này không được thỏa mãn thì có thể tạo lập chỉ dẫn mới, tất nhiên khi đó cần phải tính đến chi phí để làm công việc này.

Nhận ngày 3-6-1985

## TÀI LIỆU THAM KHẢO

- [1] Nguyễn Thanh Thủy, Một vài thuật toán cải đặt tốt nhóm các phép toán chọn—chiếu—nối tự nhiên và đánh giá độ phức tạp thời gian và bộ nhớ của chúng. Tìm hiểu hệ cơ sở dữ liệu. Viện Khoa học thống kê, Hà Nội, 1985, 59—68.
- [2] Nguyễn Thanh Thủy, Một thuật toán cải đặt tối ưu đối với phép chọn và phép tìm kiếm. Tìm hiểu hệ cơ sở dữ liệu. Viện Khoa học thống kê, Hà Nội, 1985, 42—46.
- [3] Blasgen M.W., Eswaran K. P. [1977], Storage and access in relational data bases, IBM systems Journal, Vol 16, N<sup>o</sup>4, 1977.
- [4] Adiba M., Delobel C. [1982], Bases de données et systèmes relationnels, Dunod, Paris, 1982.
- [5] Beck L. L. [1979], A generalized implementation method for relational data sublanguages. IEEE Trans. on Software engineering, vol SE-6, N<sup>o</sup>2, march, 1979.
- [6] Griffiths P. S. et al. [1979], Access path selection in a relational management systems, Proc. ACM SIGMOD conf. 1979.

## ABSTRACT

### Some modified implementation algorithms for the generalized operation PROJECTION—SELECTION—JOIN

In [1] some effective implementation algorithms for the generalized operation SELECTION—PROJECTION—JOIN and their memory and time cost estimations have been shown. In this paper, will be given some their modified implementation algorithms and their time, memory complexities.