Tạp chí Tin học và Điều khiển học, T.23, S.3 (2007), 240–250

MÔ HÌNH ƯỚC LƯỢNG CHI PHÍ XỬ LÝ TRUY VẤN ĐỐI TƯỢNG TRONG CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG

LÊ MẠNH THẠNH[1], HOÀNG BẢO HÙNG[2]

[1]*Đại học Huế*
[2]*Sở Bưu chính Viễn thông Thừa Thiên Huế,*

Abstract. In this paper, we research cost model on building blocks in query processing with basic cost factors, which we consider additional evaluation cost on clustered collection class. We used object algebra transformation rules [11] and the relational algebra expressions optimization algorithm [9], we extend them and proposed the optimization algorithm on object algebra expressions, then we illustrate this algorithm in the Object - Oriented database model ODMG along with query language OQL and its object algebra ([2, 8]).

Tóm tắt. Trong bài báo này, chúng ta nghiên cứu mô hình chi phí trên các khối dựng sẵn trong xử lý truy vấn với các yếu tố chi phí cơ sở, trong đó có sự xem xét bổ sung chi phí xử lý trên các lớp sưu tập gộp nhóm. Dựa trên các luật biến đổi đại số đối tượng trong [11] và giải thuật tối ưu hóa biểu thức đại số quan hệ trong [9], chúng ta mở rộng tập luật và xây dựng thuật toán tối ưu trên lớp các biểu thức đại số đối tượng, sau đó minh họa thuật toán trên mô hình dữ liệu hướng đối tượng ODMG với ngôn ngữ truy vấn OQL và đại số đối tượng tương ứng ([2, 8]).

1. MỞ ĐẦU

Xét quá trình thực thi truy vấn trong một hệ thống, điều chúng ta phải quan tâm là làm sao để cực tiểu tần suất sử dụng của CPU, bộ nhớ, chi phí vào/ra (I/O) và các nguồn tài nguyên về lĩnh vực truyền thông. Với kỹ thuật phần cứng hiện nay (khả năng của các chip nhớ) thì việc tối ưu thực thi một truy vấn chỉ còn là vấn đề làm cực tiểu thời gian trả lời của truy vấn, trong khi đó các hệ thống lại chịu sự chi phối chủ yếu ở thời gian trao đổi vào/ra. Mặt khác, một vấn đề cần quan tâm đó là tính đúng đắn của truy vấn, các truy vấn cho trả lời nhanh nhất nhưng nếu là câu trả lời sai hay không có được câu trả lời thì vẫn không hiệu quả.

Trong những năm gần đây vấn đề tối ưu hóa truy vấn hướng đối tượng được nhiều nhà nghiên cứu quan tâm, các kỹ thuật tối ưu hóa truy vấn được phát triển có tính kế thừa từ mô hình CSDL quan hệ như tối ưu hóa trên các biểu thức đại số đối tượng [6] và mở rộng các kỹ thuật tối ưu hóa truy vấn khác đã có trên mô hình quan hệ [9].

Tuy nhiên, kỹ thuật tối ưu hóa truy vấn trên CSDL hướng đối tượng có những điểm khác biệt so với các phương pháp tối ưu hóa truy vấn trên CSDL quan hệ - điều này xuất phát từ ngữ nghĩa của mô hình dữ liệu hướng đối tượng và các ngôn ngữ truy vấn trên mô hình này, vì vậy nhiều tác giả đã nghiên cứu và đề xuất các kỹ thuật tối ưu hóa truy vấn đối tượng thích hợp cho mô hình CSDL hướng đối tượng. Fung C.W [4] đã phát triển mô hình phân

tích chi phí đối với việc xử lý truy vấn trên các lớp không phân mảnh và các lớp phân mảnh dọc gồm cả 2 quan hệ phân cấp - phân cấp lớp hợp thành và phân cấp lớp con, nhưng chưa xem xét chi phí xử lý truy vấn trên các lớp sưu tập tụ nhóm. Gardarin [5] mở rộng mô hình chi phí xử lý truy vấn đối với các lớp sưu tập tụ nhóm, từ các kết quả này, chúng ta phát biểu lại định lý Yao [10] trong trường hợp tổng quát với các lớp sưu tập tụ nhóm và kích thước của đối tượng lớn hơn kích thước của trang bộ nhớ. Với các luật biến đổi biểu thức đại số đối tượng trong [11] chúng ta mở rộng giải thuật tối ưu hóa biểu thức đại số quan hệ trong [9] cho lớp các biểu thức đại số đối tượng và minh họa giải thuật trên mô hình CSDL hướng đối tượng ODMG với ngôn ngữ truy vấn OQL và đại số đối tượng của nó ([2,8]).

1.1. Các định nghĩa mở đầu

Định nghĩa 1.1. Một tân từ đơn giản là một tân từ định nghĩa trên một thuộc tính đơn giản hoặc trên một phương thức đơn giản và nó được xác định như sau

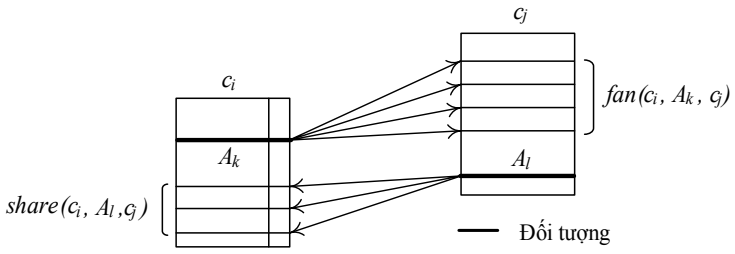
⟨Thuộc_tính⟩/ ⟨Phương_thức⟩ ⟨Phép_toán⟩ ⟨Giá_trị⟩

trong đó, ⟨Phép_toán⟩ là phép so sánh $(=, <, \leqslant, >, \geqslant, \neq)$, ⟨Giá_trị⟩ được chọn từ miền giá trị của ⟨Thuộc_tính⟩ hoặc là giá trị trả về của ⟨Phương_thức⟩.

Định nghĩa 1.2. Một đường dẫn P biểu diễn một nhánh trong phân cấp lớp hợp thành và nó được mô tả như sau: $P : c_1.A_1.A_2...A_n$ $(n \geqslant 1)$, với c_1 là một lớp trong lược đồ hướng đối tượng, A_1 là thuộc tính của lớp c_1 và A_i là thuộc tính của lớp c_i sao cho c_i là miền giá trị của thuộc tính A_{i-1} của lớp c_{i-1}, $(1 < i \leqslant n)$. Đối với lớp cuối cùng c_n trong đường dẫn, thuộc tính A_n hoặc phương thức m_n của lớp này trả về một tập các giá trị hoặc tập các định danh đối tượng (OID).

Chiều dài của đường dẫn P được định nghĩa bằng số các thuộc tính n có trong P. Chúng ta gọi lớp c_1 là lớp khởi đầu và thuộc tính cuối cùng (hoặc phương thức) A_n là thuộc tính/phương thức kết thúc của đường dẫn P.

Biểu thức điều kiện kết nối ẩn có dạng $c_1.A_1.A_2...A_n.v$ ⟨Phép_toán⟩ const, với ⟨Phép_toán⟩ là phép toán so sánh $(=, <, \leqslant, >, \geqslant, \neq)$, const là giá trị hằng trong miền của biến thể hiện dựa trên giá trị v của lớp c_n.



Hình 1. $fan(c_i, A_k, c_j)$ và $share(c_i, A_k, c_k)$

Định nghĩa 1.3. ([3]) Cho hai lớp c_i và c_j trong một đường dẫn P (c_j là miền giá trị của thuộc tính A_k của c_i). Chúng ta định nghĩa hệ số phân đầu ra, ký hiệu $fan(c_i, A_k, c_j)$ là giá trị trung bình của số các đối tượng c_j tham chiếu đến một đối tượng của c_i thông qua thuộc tính A_k. Tương tự, mức chia sẻ, ký hiệu $share(c_i, A_l, c_k)$ là giá trị trung bình của số các đối tượng c_i tham chiếu đến cùng đối tượng của c_j thông qua thuộc tính A_l. Chúng ta ký hiệu $FAN(c_i, c_j)$ và $SHARE(c_i, c_j)$ là giá trị trung bình của $fan(c_i, A_k, c_j)$ và $share(c_i, A_l, c_j)$

trên tất cả các đối tượng của lớp c_j và lớp c_i.

1.2. Các đóng góp và tổ chức của bài báo

Những đóng góp chính của bài báo bao gồm:

+ Nghiên cứu, phân tích mô hình chi phí cơ sở có xem xét bổ sung chi phí xử lý đối với các lớp sưu tập gộp nhóm trong trường hợp kích thước của đối tượng lớn hơn kích thước của trang bộ nhớ.
+ Mở rộng tập luật biến đổi các biểu thức đại số đối tượng và xây dựng thuật toán tối ưu các biểu thức đại số đối tượng. Minh họa giải thuật này trên mô hình dữ liệu hướng đối tượng ODMG với ngôn ngữ truy vấn đối tượng OQL.

Cấu trúc của bài báo được tổ chức như sau: Mục 1, mở đầu giới thiệu một số khái niệm cơ sở phục vụ cho mô hình chi phí cơ sở trên các khối dựng sẵn ở Mục 2. Mục 3, mở rộng các luật biến đổi đại số đối tượng, là cơ sở để xây dựng thuật toán tối ưu hóa truy vấn bằng phương pháp biến đổi các biểu thức đại số đối tượng và cuối cùng là kết luận, hướng phát triển của bài báo.

2. MÔ HÌNH CHI PHÍ VỚI CÁC KHỐI DỰNG SẴN

Phần này sẽ giới thiệu mô hình phân tích chi phí tổng quan đối với việc xử lý truy vấn trên các lớp ([4]).

Bảng 1. Bảng các tham số của mô hình chi phí

Phạm vi	Tham số	Ý nghĩa
CSDL	$\|\|c_{i,k}\|\|$	Lực lượng của lớp $c_{i,k}$ (tức là lớp con thứ k của lớp thứ i theo phân cấp lớp hợp thành)
	$\|c_{i,k}\|$	Số các trang sử dụng cho lớp $c_{i,k}$
	$SC_{i,k}$	Kích thước của đối tượng (tính bằng byte) của lớp $c_{i,k}$
	q_i	Số các lớp con trong phân cấp lớp con có gốc là lớp c_i
	$fan_{i-1,j,i,k}$	*fan-out* đối với phân cấp lớp hợp thành từ lớp con thứ j của lớp c_{i-1} với lớp con thứ k của lớp c_i
	l_p	Chiều dài đường dẫn của biểu thức đường dẫn, nghĩa là, số các lớp thuộc về biểu thức đường dẫn trong phân cấp lớp hợp thành.
	$NP_{i,k}$	Số các đối tượng (của lớp con thứ k của lớp c_i) trên mỗi trang. Nếu $SC_{i,k} < PS$ thì $NP_{i,k}$ là $\left\lfloor \frac{PS}{SC_{i,k}} \right\rfloor$, ngoài ra nó có giá trị là 1.
	b	Hệ số phân đầu ra *fan* trung bình của chỉ mục B^+-cây
	PS	Kích thước trang của hệ thống file (đơn vị là byte).
Truy vấn	$ref_{i,k}$	Số các tham chiếu đối tượng đối với lớp con thứ k trong lớp c_i trong tiến trình định giá biểu thức đường dẫn theo phân cấp lớp hợp thành.
	SEL_i	Số các đối tượng được chọn của truy vấn theo tân từ trên lớp c_i

Tổng chi phí của tiến trình xử lý truy vấn được cho bởi công thức

$$Total\_cost = IO\_cost + CPU\_cost$$

IO_cost là chi phí của vào/ra của đĩa và *CPU_cost* là chi phí tính toán suốt tiến trình xử lý truy vấn. Trong đó, chúng ta tập trung nghiên cứu trên chi phí *IO_cost* và bỏ qua chi phí *CPU_cost*. Điều này sở dĩ như vậy là vì, với mỗi CSDL ứng dụng rất lớn với số lượng khổng lồ các truy xuất dữ liệu thì sự tham gia của chi phí *CPU_cost* vào tổng chi phí *Total_cost* sẽ không đáng kể.

Tương tự phương pháp trong [9], chúng ta sẽ xem xét sự khác biệt về số lần truy xuất đĩa do ảnh hưởng của dung lượng của bộ nhớ chính đối với việc xử lý truy vấn:

(1) Giả thiết bộ nhớ lớn: kích cở bộ nhớ chính là đủ lớn để chúng ta có đủ các vùng nhớ đệm cho việc nạp vào tất cả các đối tượng (các đối tượng được nạp vào từ đĩa và chỉ nạp 1 lần).

(2) Giả thiết bộ nhớ nhỏ: kích cở bộ nhớ chính quá nhỏ, chúng ta chỉ có thể cấp một trang bộ nhớ đệm cho mỗi lớp (các đối tượng sẽ được nạp vào bộ nhớ chính trong nhiều lần và điều này làm tăng cao số lần truy xuất đĩa).

Mô hình chi phí của chúng ta dựa trên một tập các tham số được phân thành hai nhóm biểu diễn trong Bảng 1, đó là các tham số hệ thống CSDL, tham số xử lý truy vấn.

2.1. Ước lượng số các trang truy xuất cho một sưu tập lớp

Tổng số trang sử dụng cho một lớp c với kích thước đối tượng SC và lực lượng $\|c\|$ được cho bởi công thức $|c| = \left\lceil \dfrac{\|c\|SC}{PS} \right\rceil$, PS là kích thước trang được dùng cho hệ thống CSDL hướng đối tượng. Khi áp dụng công thức này đối với phân cấp lớp con, chúng ta giả sử rằng các đối tượng của cùng lớp (lớp con) được lưu trữ cùng nhau, nhưng giữa các lớp (lớp con) khác nhau các đối tượng được lưu trữ tách rời nhau. Điều này có nghĩa rằng tất cả các lớp con sẽ không được nhóm lại trong một lớp sưu tập lớn với mục đích là hiệu quả mang lại của việc xử lý trên các lớp con riêng lẻ. Cùng giả thiết như vậy đối với việc lưu trữ các đối tượng hợp thành.

2.2. Ước lượng số trang truy xuất để chọn một số đối tượng

Định lý 2.1. (Định lý Yao) ([10]) *Cho n bản ghi cùng kiểu được nhóm vào m trang ($1 < m \leqslant n$), mỗi trang chứa n/m bản ghi. Nếu k bản ghi ($k \leqslant n - n/m$) được chọn một cách ngẫu nhiên từ n bản ghi, thì số các trang truy xuất là*

$$Yao(n, m, k) = m \left[1 - \prod_{i=1}^{k} \frac{nd - i + 1}{n - i + 1} \right], \tag{1}$$

với $d = 1 - \dfrac{1}{m}$.

Số trang truy xuất không bằng k bởi vì một số trang có thể chứa hai hoặc nhiều hơn các bản ghi kết quả. Áp dụng hàm Yao trong CSDL hướng đối tượng, chúng ta lấy $n = \|c\|$, $m = |c|$ và $k = SEL\|c\|$ (SEL là số các đối tượng được chọn theo điều kiện trên lớp hiện thời c).

Mặt khác, nếu c là một sưu tập gộp nhóm, chúng ta không thể sử dụng hàm Yao để ước lượng các trang truy xuất bởi công thức $Yao(\|c\|, |c|, SEL\|c\|)$ thường được sử dụng trong các hệ quản trị CSDL quan hệ, vì sưu tập gộp nhóm c có nhiều hơn một phân mảnh. Các đối tượng được gộp nhóm với nhau từ các sưu tập khác nhau, trong khi có một số đối tượng

được lưu trữ đơn lẻ. Vì vậy, mật độ của các đối tượng trong các phân mảnh khác nhau là không bằng nhau. Trong trường hợp này chúng ta sử dụng công thức ước lượng trong [5] được phát biểu theo định lý sau.

Định lý 2.2. (Định lý Yao') ([5]) *Cho sưu tập c có p phân mảnh, mỗi phân mảnh có $\|c_i\|$ đối tượng. Nếu k đối tượng ($k < \sum_{i=1}^{p} \|c_i\|$) được chọn từ c một cách ngẫu nhiên thì số các trang được truy xuất là*

$$Yao'(c,k) = \sum_{i=1}^{p} Yao(\|c_i\|, |c_i|, k_i) \tag{2}$$

với k_i là số các đối tượng được chọn trong phân mảnh c_i.

Nếu các đối tượng chọn được lấy giống nhau giữa các phân mảnh thì $k_i = \dfrac{\|c_i\|}{\|c\|}k$. Ngược lại, nếu các đối tượng thoả mãn tân từ không được lấy giống nhau giữa các phân mảnh khác nhau thì chúng ta phải dùng sự lựa chọn trên mỗi phân mảnh để xác định chính xác giá trị của mỗi k_i. Công thức (2) tổng quát hơn công thức (1) và công thức (1) là một trường hợp đặc biệt của (2) khi sưu tập c chỉ có một phân mảnh.

Công thức (1) chỉ áp dụng được khi $m \leqslant n$, tức là khi kích thước đối tượng là nhỏ hơn hoặc bằng kích thước của trang. Đối với kích thước đối tượng lớn hơn kích thước trang, chúng ta ước lượng số các trang bằng $\sum_{i=1}^{p} |c_i| \dfrac{k}{\sum_{i=1}^{p} \|c_i\|}$. Vì vậy, trong mô hình chi phí các khối dựng sẵn, chúng ta xây dựng lại hàm Y như sau

$$Y(c,k) = \begin{cases} \sum_{i=1}^{p} Yao(\|c_i\|, |c_i|, k_i) & \text{Với kích thước đối tượng} \leqslant \text{ kích thước trang} \\ \sum_{i=1}^{p} |c_i| \dfrac{k}{\sum_{i=1}^{p} \|c_i\|} & \text{Với kích thước đối tượng} > \text{ kích thước trang} \end{cases} \tag{1}$$

Như vậy, với việc xây dựng hàm $Y()$ và công thức (2), Định lý 2.2 được phát biểu lại như sau.

Định lý 2.3. *Cho sưu tập c có p phân mảnh, mỗi phân mảnh có $\|c_i\|$ đối tượng. Nếu k đối tượng ($k < \sum_{i=1}^{p} \|c_i\|$) được chọn từ c một cách ngẫu nhiên thì số các trang được truy xuất là $Yao'(c,k) = Y(c,k)$, với k_i là số các đối tượng được chọn trong phân mảnh c_i.*

Chứng minh. Định lý 2.3 được suy trực tiếp từ Định lý 2.1 và 2.2, trong đó hàm $Y'(C,k)$ được mở rộng cho trường hợp kích thước của đối tượng lớn hơn kích thước của trang bộ nhớ (hàm $Y(c,k)$). ■

2.3. Ước lượng số các trang truy xuất với chỉ mục tìm kiếm

Nếu tân từ trong truy vấn chứa một biến thể hiện kết hợp với một chỉ mục, chúng ta sử dụng chỉ mục này để giải quyết việc nạp các đối tượng lớp gốc. Với chỉ mục tụ nhóm B^+-cây và giá trị trung bình fan là b [5, 9], số các trang truy xuất được yêu cầu là

$\log_b\left(SEL \times \sum_{k=0}^{q_i} \frac{||c_{1,k}||}{NP_{1,k}}\right)$ để tìm kiếm theo chỉ mục tụ nhóm, q_i là số các lớp con trong phân cấp lớp con có gốc là lớp c_i, SEL là số các đối tượng được chọn theo tân từ trên lớp gốc và $NP_{1,k}$ là số các đối tượng (của lớp con thứ k của lớp gốc trong mỗi trang). Đối với chỉ mục không tụ nhóm ([5, 9]), số các trang truy xuất yêu cầu là $\log_b\left(SEL \times \sum_{k=0}^{q_i} ||c_{1,k}||\right)$.

2.4. Ước lượng số các tham chiếu đối tượng

Cần ước lượng số các tham chiếu đối tượng trong lúc định giá tân từ (cùng với phân cấp lớp hợp thành).

Duyệt tuần tự, $ref_{1,k} = ||c_{1,k}||,\ 0 \leqslant k \leqslant q_1$ và

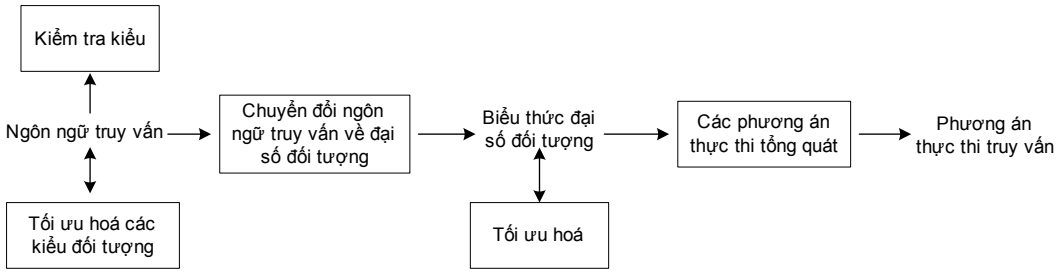
$ref_{1,k} = \left(\sum_{j=0}^{q_{i-1}} ref_{i-1,j} \times fan_{i-1,i,j,k}\right),\ 1 < i \leqslant l_p$ và $0 \leqslant k \leqslant q_i$.

Duyệt với chỉ mục tụ nhóm, $ref_{1,k} = SEL_1 \times ||c_{1,k}||,\ 0 \leqslant k \leqslant q_1$,

$ref_{2,k} = \left(\sum_{j=0}^{q_{i-1}} ref_{1,j} \times fan_{1,2,j,k}\right) \times SEL_{i-1},\ 0 < k \leqslant q_2$ và

$ref_{i,k} = \left(\sum_{j=0}^{q_{i-1}} ref_{i-1,j} \times fan_{i-1,1,j,k}\right) \times SEL_{i-1},\ 2 < i \leqslant q_2$. Duyệt với chỉ mục không tụ nhóm, $ref_{1,k}$ có công thức giống như duyệt chỉ mục tụ nhóm.

3. TỐI ƯU HÓA TRUY VẤN ĐỐI TƯỢNG BẰNG CÁC PHÉP BIẾN ĐỔI BIỂU THỨC ĐẠI SỐ ĐỐI TƯỢNG



Hình 2. Tiến trình khung xử lý truy vấn

Tiến trình tổng quát tối ưu hóa truy vấn đối tượng dựa trên tập luật được mô tả trong Hình 2, đầu vào của tiến trình xử lý là các truy vấn được viết bằng ngôn ngữ truy vấn đối tượng, chuyển đổi các truy vấn thành các biểu thức đại số đối tượng tương đương. Sau đó, áp dụng các luật biến đổi trên các phép toán đại số như chọn, chiếu, kết nối với các lớp sưu tập, loại bỏ trùng lặp trong các đa tập,... Cuối cùng, chúng ta có kết quả là phương án thực thi được chọn trong tiến trình tối ưu truy vấn.

3.1. Sự biểu diễn tương đương giữa truy vấn OQL và đại số đối tượng

Để đảm bảo cho quá trình chuyển đổi các biểu thức đại số đối tượng thành truy vấn OQL và ngược lại, là không làm nảy sinh dư thừa dữ liệu khi thực thi truy vấn. Chúng ta chứng

minh rằng sự chuyển đổi giữa truy vấn OQL và biểu thức đại số đối tượng là tương đương. Các kết quả của Định lý 3.1, 3.2 trong [1] đã chứng minh điều này.

Định lý 3.1. [1] *Mọi biểu thức đại số đối tượng đều biểu diễn được bằng các truy vấn đối tượng trong OQL.*

Định lý 3.2. [1] *Mọi truy vấn đối tượng trong OQL đều biểu diễn được bằng các biểu thức đại số đối tượng.*

Như vậy, việc viết lại một truy vấn đã cho các thành biểu thức đại số bằng tập phép toán đại số đối tượng là tương đương. Các biểu thức đại số này có thể được ước lượng với các chi phí xử lý khác nhau. Vì vậy, về mặt lý thuyết chúng ta mong muốn tìm được các biểu thức đại số tương đương với một truy vấn sao cho có thể đạt được một phương án thực thi hiệu quả hơn. Tuy nhiên, về mặt cài đặt, vì số lượng các truy vấn tương đương quá lớn, trong lúc đó chúng ta chỉ cần một tập con các truy vấn này mà thôi. Do đó, để tìm ra các truy vấn tương đương khác, chúng ta sẽ cần một tập luật biến đổi bảo toàn tương đương, nhưng mô hình dữ liệu hướng đối tượng lại không có một đại số đối tượng chuẩn áp dụng được cho tất cả các mô hình hướng đối tượng, cho nên sự kỳ vọng để có một tập chuẩn tắc gồm các luật biến đổi bảo toàn tương đương là không tồn tại. Vì vậy, chúng ta chỉ mong muốn chứng tỏ rằng sự biến đổi bảo toàn tương đương trên cơ sở đại số đối tượng là đúng, với một số luật biến đổi được trình bày sau đây.

3.2. Các luật biến đổi đại số đối tượng

Ký hiệu S, S_1, S_2, S_3 là các tập hợp phần tử; f, g, h là các biểu thức điều kiện,e là biểu thức đại số, phép toán $op \in \{union, diff\}$. Những luật này chỉ áp dụng trên các phép toán đối tượng, phép toán bộ, các phép toán tập hợp và phép toán về kiểu dữ liệu "túi". Về mặt ký hiệu chúng ta chỉ sử dụng các ký hiệu phép toán một cách hình thức, các phép toán này có thể được cài đặt với một số thay đổi trong các mô hình khác nhau ([8, 11]).

L1. Giao hoán phép chọn: $\sigma_{\lambda t.g}(\sigma_{\lambda s.f}(S)) = \sigma_{\lambda s.f}(\sigma_{\lambda t.g}(S))$

L2. Tổ hợp các phép chọn $\sigma_{\lambda.s(f\wedge g\wedge ...\wedge h}(S) = \sigma_{\lambda.s.f}(\sigma_{\lambda t.g}(...(\sigma_{\lambda u.h}(S))...))$

L3. Dãy các phép chiếu

$$\pi_{(a_1,...,a_n)}(\pi_{b_1,...,b_m}(S)) = \pi_{(a_1,...,a_n)}(S), \quad \text{với } \{(a_1, ..., a_n)\} \subset \{b_1, ..., b_m\}$$

L4. Giao hoán phép chọn và phép chiếu

$$\sigma_{\lambda s.e}(\pi_{(a_1,...,a_n)}(S)) = \pi_{(a_1,...,a_n)}(\sigma_{\lambda s.e}(S))$$

L5. Giao hoán một phép chiếu với phép hợp, hiệu trên tập/đa tập

$$\pi_{(a_1,...,a_n)}(S_1 \ op \ S_2) = \pi_{(a_1,...,a_n)}(S_1) \ op \ \pi_{(a_1,...,a_n)}(S_2)$$

L6. Phân phối phép chọn với phép hợp và phép hiệu trên tập/đa tập

$$\sigma_{\lambda s.f}(S_1 \ op \ S_2) = \sigma_{\lambda s.f}(S_1) \ op \ S_2, \quad \text{nếu} f \text{ chỉ liên quan với} S_1.$$

Tổng quát: $\sigma_{\lambda s.(f\wedge g\wedge h)}(S_1 \ op \ S_2) = \sigma_{\lambda u.h}(\sigma_{\lambda s.f}(S_1) \ op \ \sigma_{\lambda t.g}(S_2))$, nếu f liên quan S_1, g, liên quan S_2 và h liên quan cả S_1 và S_2.

L7. Giao hoán giữa phép apply và phép chọn: nếu điều kiện chọn chỉ chứa các thuộc tính do phép toán apply trả về thì

$$apply_{\lambda s.e}(\sigma_{\lambda t.f}(S)) = \sigma_{\lambda t.f}(apply_{\lambda s.e}(S))$$

L8. Giao hoán giữa phép làm phẳng (flat) và phép apply trên tập/đa tập. Giả sử S là thể hiện của một lớp và X là một tập thuộc tính phức của lớp.

$$flat(apply_{\lambda s.(apply_{\lambda t.e}(\pi(X)(\pi_V(S))))}(S)) = apply_{\lambda t.e}(flat(apply_{\lambda s.\pi(X)(\pi_V(S))}(S)))$$

Biểu thức ở vế trái, có biểu thức e tác động trước tập các tập (thu được bởi π_X) sau đó làm phẳng thành một tập. Biểu thức ở vế phải có phép toán làm phẳng được tác động trước (kết quả thu được là một tập), sau đó thực hiện phép toán *apply*.

L9. Tính kết hợp của phép hợp: $(S_1\ union\ S_2)\ union\ S_3 = S_1\ union\ (S_2\ union\ S_3)$

L10. Các luật kế thừa đối với phép chọn và phép apply: nếu S_2 là một lớp con của S_1, thì thể hiện của S_2 là một tập con của thể hiện của S_1

$$\sigma_{\lambda s.f}(S_1)\ union\ \sigma_{\lambda s.f}(S_2) = \sigma_{\lambda s.f}(S_1)$$

$$apply_{\lambda s.e}(S_1)\ union\ apply_{\lambda s.e}(S_2) = apply_{\lambda s.e}(S_1)$$

3.3. Thuật toán tối ưu hóa các biểu thức đại số đối tượng

Chúng ta áp dụng các luật trong Mục 3.2 để thực hiện ước lượng các biểu thức đại số đối tượng. Thuật toán sẽ tập trung xử lý các phép toán chiếu, chọn, áp dụng biểu thức đại số (set_apply) trên các kiểu đối tượng và phép toán loại bỏ trùng lặp trên các đa tập, lớp sưu tập.

Thuật toán. Tối ưu hóa các biểu thức đại số đối tượng dựa trên tập luật.
Vào. Biểu thức đại số đối tượng.
Ra. Một dãy các bước ước lượng biểu thức đại số đối tượng.
Phương pháp.
(1) Khởi tạo cây phân tích cú pháp từ biểu thức đại số đối tượng.
(2) Sử dụng luật (L2) tách phép chọn $\sigma_{\lambda s.(f \wedge g \wedge ... \wedge h)}(S)$ thành chuỗi các phép chọn

$$\sigma_{\lambda s.f}(\sigma_{\lambda t.g}(...(\sigma_{\lambda u.h}(S))...))$$

(3) Sử dụng các luật kế thừa đối với các phép chiếu (L3), phép chọn và phép apply (L10) tổ hợp dãy các phép chiếu, chọn thành một phép chiếu và một phép chọn.
(4) Đối với mỗi phép chọn, sử dụng các luật (L4, L6, L7, L10) "đẩy" các phép chọn xuống các lớp thành phần hoặc "qua" các nút kết nối và phép tạo nhóm.
(5) Đối với mỗi phép chiếu (đối tượng, tập, bộ), sử dụng luật (L3, L4, L5) để di chuyển phép chiếu xuống càng sâu càng tốt. Nếu tập thuộc tính được chiếu bao gồm tất cả các thuộc tính của biểu thức thì chúng ta loại bỏ phép chiếu đó.
(6) Sử dụng các luật (L8, L9, L10) trên các lớp sưu tập, để loại bỏ các phần tử trùng lặp trong các lớp sưu tập; di chuyển phép làm phẳng (flat), phép loại bỏ trùng lặp trong các đa tập (bagtoset) lên trước các phép toán nhóm hoặc kết nối.

(7) Tạo ra dãy các bước biến đổi để ước lượng mỗi biểu thức đại số theo một thứ tự sao cho không có biểu thức nào được ước lượng trước các biểu thức con của nó.

Mệnh đề 3.3. *Thuật toán tối ưu hóa các biểu thức đại số dựa trên tập luật là đúng đắn và kết quả của thuật toán là phương án truy vấn có chi phí thấp hơn chi phí ước lượng của biểu thức đầu vào thuật toán.*

Chứng minh.

Theo Định lý 3.1, 3.2 và tập các luật biến đổi biểu thức đại số đối tượng, chúng ta suy ra các bước thực thi trong thuật toán cho kết quả đúng và tương đương.

Áp dụng các phép toán chiếu trên các lớp đối tượng làm giảm kích thước các lớp và các đối tượng tham gia trong biểu thức, điều này sẽ làm giảm chi phí nạp lớp vào bộ nhớ trong (IO_Load). Mặt khác, phép toán làm phẳng (set_flat) và loại bỏ trùng lặp (bagtoset) áp dụng cho đa tập, lớp sưu tập sẽ làm giảm một cách đáng kể các biến thể của các lớp tham gia trong các phép kết nối, nhóm tương đương (IO_Eval). (

Độ phức tạp tính toán của thuật toán có thời gian đa thức theo kích thước (số các biến thể) của các lớp tham gia trong biểu thức.

3.4. Ví dụ minh họa

Chúng ta xét các truy vấn và chuyển đổi biểu thức đại số đối tượng tương ứng, sau đó áp dụng thuật toán trong Mục 3.3 thực hiện tối ưu hóa trên cây phân tích cú pháp. Ngôn ngữ truy vấn được sử dụng là ngôn ngữ OQL được xây dựng trên mô hình dữ liệu hướng đối tượng ODMG.

Ví dụ. Cho lược đồ đối tượng VienDaihoc được định nghĩa trong OQL như sau:

class NhanSu

type tuple (maso: int, hoten: string, pho: string, tpho: string, matinh: int,

ngaysinh: tuple (ngay: int, thang: int, nam:int))

class SinhVien inherits NhanSu

type tuple (gvhd: string, dtb: float, hocbong: float, tenkhoa: Khoa)

class GiangVien inherits NhanSu

type tuple (bomon: string, mabomon: int, chucvu: string, tenkhoa: Khoa, luong: int, con: set(NhanSu))

class Khoa

type tuple (makhoa: int, tenkh: string, diadiem: string, ngansach: float, cbgd: set(GiangVien))

Chúng ta xét một số các truy vấn minh họa trong OQL và cây phân tích cú pháp của biểu thức đại số đối tượng tương ứng:

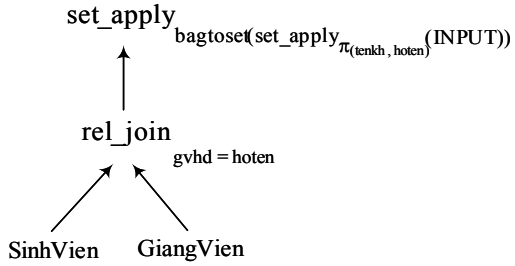
Truy vấn 1: Ta có truy vấn

define SinhVien as s, GiangVien as e

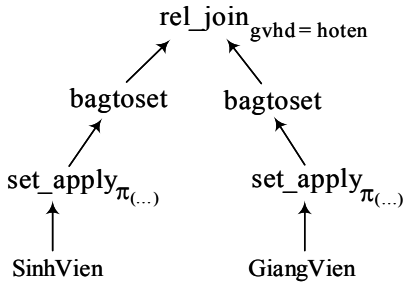
select distinct (s.tenkhoa.tenkh, e.hoten)

where s.gvhd = e.hoten

Hình 3.2 biểu diễn truy vấn 1 bằng cây phân tích cú pháp.



Hình 3.2 Khởi tạo cây phân tích cú pháp



Hình 3.3. Cây biến đổi của Hình 3.2

Hình 3.3 biểu diễn cây phân tích cú pháp khi chúng ta áp dụng luật "đẩy" phép toán bagtoset lên trước toán tử kết nối rel_join, luật này áp dụng khi các thành phần trùng lặp quá lớn tồn tại trong các đa tập, như vậy toán tử bagtoset chỉ thực hiện trên $|s| + |e|$ biến thể (trong trường hợp xấu nhất) tốt hơn $|s| * |e|$ biến thể (đối với trường hợp phép kết nối được thực hiện trước). Và tiếp tục đẩy toán tử (qua nút "join".

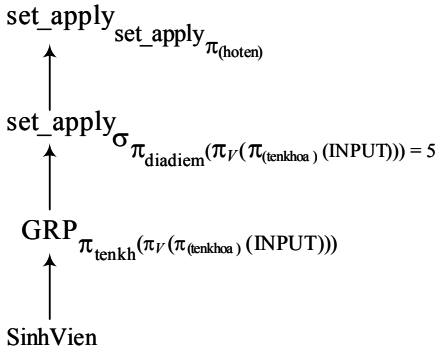
Truy vấn 2: Tìm tên các sinh viên của khoa có văn phòng khoa đặt ở tầng 5 (thuộc tính diadiem). Tên các sinh viên được nhóm theo thuộc tính tenkh (phân loại khoa, ví dụ Sinh học, Công nghệ thông tin,...):

define SinhVien as s

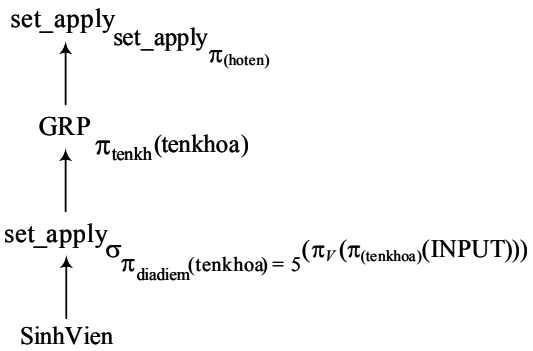
select (s.hoten)

group by s.tenkhoa.tenkh

where s.tenkhoa.diadiem = 5



Hình 3.4. Khởi tạo cây



Hình 3.5. Cây kết quả sau khi áp dụng các luật chuyển đổi

Hình 3.4 biểu diễn cây phân tích cú pháp đại số cho truy vấn 2, ta nhóm đa tập trên thuộc tính tenkh của thuộc tính tenkhoa, sau đó loại bỏ các sinh viên của các khoa không ở tầng 5, cuối cùng chiếu lấy thuộc tính hoten.

Một phương pháp tối ưu truy vấn 2 được suy trực tiếp từ Hình 3.4. Trước hết, ta đưa phép chọn lên trước phép tạo nhóm GRP và sử dụng các phép chiếu trên đối tượng (π_V) để loại bỏ tham chiếu tên thuộc tính tenkhoa, sau đó trích chiếu lấy giá trị của thuộc tính diadiem.

4. KẾT LUẬN

Mô hình ước lượng chi phí giới thiệu trong bài báo được xem xét trên cơ sở một số yếu tố chi phí cơ bản. Trong khuôn khổ của bài báo, chúng ta chưa đi sâu phân tích các trường hợp tổng quát của mô hình ước lượng chi phí xử lý truy vấn hướng đối tượng trên các thành phần như chi phí nạp các đối tượng vào bộ nhớ chính, chi phí ước lượng biểu thức đường dẫn và chi phí kết xuất kết quả của truy vấn. Chúng ta sẽ tiếp tục nghiên cứu chi tiết mô hình ước lượng chi phí xử lý truy vấn đối tượng, để từ đó thể nghiệm, so sánh và đánh giá tính hiệu quả của thuật toán tối ưu hóa biểu thức đại số đối tượng. Mặt khác, dựa vào mô hình ước lượng chi phí xử lý truy vấn, chúng ta xem xét tính ưu tiên về thứ tự thực hiện của các phép toán đại số đối tượng trong tiến trình thực thi truy vấn đối tượng.

TÀI LIỆU THAM KHẢO

[1] Đoàn Văn Ban, Lê Mạnh Thạnh, Hoàng Bảo Hùng, Sự tương đương trong biểu diễn giữa ngôn ngữ truy vấn OQL và đại số đối tượng, *Tạp chí Tin học và Điều khiển học* **20** (3) (2004) 257–269.

[2] R. G. G. Cattel, D. K. Barry, *The Object Database Standard: ODMG 3.0*, Morgan Kaufmann Publishers, 2000.

[3] W. S. Cho, C. M. Park, K. Y. Whang, and S. H. So, A new method for estimating the number of objects satisfying an object-oriented query involving partial participation of classes, *Information Systems* **21** (3) (1996) 253–267.

[4] C. W. Fung, K. Karlapalem, and Q. Li, An evaluation of vertical class partitioning for query processing in object-oriented databases, *IEEE Transactions on Knowledge and Data Engineering* **14** (5) (2002) 1095–1118.

[5] G. Gardarin, J. R. Gruser, and Z. H. Tang, A cost model for clustered object-oriented databases, *Proceedings of the 21st VLDB Conference*, Switzerland, 1995 (323–334).

[6] Lê Mạnh Thạnh, Hoàng Bảo Hùng, Ngôn ngữ truy vấn hướng đối tượng và tối ưu hóa truy vấn trên CSDL hướng đối tượng bằng phương pháp biến đổi đại số, "Kỷ yếu Hội nghị khoa học kỷ niệm 25 năm thành lập Viện Công nghệ thông tin", Hà nội, 12/2001.

[7] Lê Mạnh Thạnh, Đoàn Văn Ban, Hoàng Bảo Hùng, Phương pháp ước lượng các truy vấn lồng trong cơ sở dữ liệu hướng đối tượng bằng siêu đồ thị kết nối, *Chuyên san Tạp chí Bưu chính Viễn thông và Công nghệ thông tin*, "Các công trình nghiên cứu - triển khai Viễn thông và Công nghệ thông tin", 14, 2005, (43–49).

[8] A. Trigoni, Semantic Optimization of OQL Queries, Technical Report, Number 547, University of Cambridge, Computer Laboratory, UCAM-CL-TR-547, ISSN 1476-2986, 2002.

[9] Ullman, D. Jeffrey, *Nguyên lý các hệ cơ sở dữ liệu và cơ sở tri thức*, Tập 2, Trần Đức Quang biên dịch, Nhà xuất bản Thống kê, 1999.

[10] S. B. Yao, Approximating block accesses in database organizations, *Communications of the ACM* **20** (4) (1977) 260–261.

[11] Yu, T. Clement, Meng, Weiyi, *Principles of Database Query Processing for Advanced Applications*, Morgan Kaufmann Publishers, Inc, California, 1998.

Nhận bài ngày 6 - 7 - 2006
Nhận lại sau sửa ngày 18 - 8 -2006