

THUẬT TOÁN PHÂN MẢNH ĐỌC VÀ CẤP PHÁT TRONG CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG PHÂN TÁN

Mai Thúy Nga^{1,*}, Đoàn Văn Ban², Nguyễn Mạnh Hùng³

¹*Khoa Toán Tin, Đại học Thăng Long, đường Nghiêm Xuân Yêm, Hoàng Mai, Hà nội*

²*Viện Công nghệ Thông tin, 18 đường Hoàng Quốc Việt, Cầu Giấy, Hà nội*

³*Học viện Kỹ thuật Quân sự, 236 đường Hoàng Quốc Việt, Cầu Giấy, Hà nội*

*Email: *mai_nga@yahoo.com*

TÓM TẮT

Các hệ thống thông tin ngày nay thường được yêu cầu xây dựng dựa trên những kiến trúc hướng đối tượng và phân tán trên mạng, bao gồm từ nhiều cơ sở tính toán đa dạng, không thuần nhất. Trong các môi trường như thế, hệ thống có sở dữ liệu hướng đối tượng phân tán cung cấp những cơ chế rất tin cậy và hiệu quả để tổ chức lưu trữ, xử lý và truy vấn khối lượng rất lớn các đối tượng khác nhau. Với các đặc tính cơ bản của hướng đối tượng như đóng gói, kế thừa, phân cấp lớp, bài toán thiết kế trong cơ sở dữ liệu hướng đối tượng phân tán phát sinh thêm nhiều vấn đề phức tạp so với bài toán thiết kế trong cơ sở dữ liệu quan hệ. Thiết kế trong cơ sở dữ liệu hướng đối tượng phân tán được chia thành hai giai đoạn: Phân mảnh lớp đối tượng và cấp phát lớp. Mục tiêu của phân mảnh và cấp phát là nâng cao hiệu năng xử lý và giảm việc truyền dữ liệu. Thông thường hai giai đoạn này là tách biệt, phân mảnh xong mới thực hiện cấp phát. Bài báo này đề cập tới một thuật toán trong đó việc phân mảnh và cấp phát trong cơ sở dữ liệu hướng đối tượng phân tán được thực hiện đồng thời, đối với trường hợp các thuộc tính và phương thức đều thuộc loại đơn giản.

Từ khóa: cơ sở dữ liệu hướng đối tượng, cơ sở dữ liệu phân tán, cơ sở dữ liệu hướng đối tượng phân tán, phân mảnh, cấp phát.

1. GIỚI THIỆU

Mô hình cơ sở dữ liệu (CSDL) quan hệ có những hạn chế trong việc quản lý các loại dữ liệu phức tạp như Multimedia, CAD/CAM, dữ liệu trong các hệ thống quản lý tài chính ... Đó là lí do xuất hiện của mô hình dữ liệu mới, mô hình cơ sở dữ liệu hướng đối tượng (CSDL HĐT). CSDL HĐT đã được ứng dụng trong nhiều lĩnh vực phức tạp và khắc phục được các hạn chế của mô hình CSDL quan hệ. Tuy nhiên với các đặc trưng cơ bản của công nghệ hướng đối tượng như tính đóng gói, kế thừa, phân cấp lớp, mô hình CSDL HĐT đòi hỏi các kĩ thuật mới cho quản lý dữ liệu [1].

Để đáp ứng nhu cầu của doanh nghiệp lớn với sự phân bố nhiều trạm ở các vị trí địa lý khác nhau, CSDL HĐT được phát triển trong môi trường mạng tạo thành mô hình cơ sở dữ liệu hướng đối tượng phân tán (CSDL HĐT PT). Trong CSDL HĐT PT, dữ liệu được phân bố trên một số trạm của mạng máy tính, và các ứng dụng sẽ phải truy cập, xử lý dữ liệu tại các trạm khác nhau. Thiết kế hiệu quả cho CSDL HĐT PT để cải thiện hiệu năng của hệ thống là một vấn đề được quan tâm trong nhiều nghiên cứu gần đây.

Bài toán thiết kế phân tán chia thành hai giai đoạn: phân mảnh là chia nhỏ dữ liệu thành các phần và cấp phát là định vị các phần vào các trạm tương ứng. Phân mảnh trong CSDL HĐT PT được thực hiện trên các lớp (class) với hai kỹ thuật chính là phân mảnh dọc (vertical fragmentation) và phân mảnh ngang (horizontal fragmentation). Phân mảnh dọc nhằm mục đích chia một lớp thành các phần của lớp, mỗi phần gồm một số thuộc tính và phương thức. Phân mảnh ngang chia các đối tượng trong cùng một lớp thành các phần khác nhau, mỗi phần gồm một số đối tượng. Trong báo cáo này chúng tôi tập trung vào thuật toán phân mảnh dọc.

Nhiều thuật toán cho phân mảnh dọc đã được đề xuất cho mô hình quan hệ, như các thuật toán trình bày trong [1, 2, 3]. Trong mô hình đối tượng việc phân mảnh này sinh các vấn đề phức tạp mới do các đặc tính của hướng đối tượng, đó là tính đóng gói, kế thừa, phân cấp lớp. Karlapalem và Li [4] đưa ra lược đồ phân hoạch cho CSDL HĐT. Ezeife và Barker [5] trình bày các thuật toán phân hoạch dọc cho từng trường hợp, thuộc tính đơn giản và phương thức đơn giản, thuộc tính đơn giản và phương thức phức tạp, thuộc tính phức tạp và phương thức đơn giản, thuộc tính phức tạp và phương thức phức tạp. Lee and Lim [6] đưa ra một thuật toán phân hoạch dựa trên thuộc tính. Rajan và Saravanan [7] đề nghị một thuật toán phân mảnh dọc sử dụng tác tử thông minh. Bài toán cấp phát lớp trong CSDL HĐT PT cũng đã được nghiên cứu trong [8, 9, 10].

Trong các hướng tiếp cận ở trên việc phân mảnh và cấp phát được thực hiện ở hai giai đoạn riêng biệt, phân mảnh được thực hiện trước, sau đó sẽ định vị các mảnh vào các trạm. Trong giai đoạn phân mảnh không sử dụng đến thông tin về chi phí kết nối giữa các trạm, tuy nhiên thông tin này hoàn toàn có thể ảnh hưởng để tạo ra một phương án phân mảnh hiệu quả hơn.

Trong bài báo này, chúng tôi mở rộng hướng tiếp cận của Hui Ma và Markus [11] đã áp dụng cho mô hình CSDL quan hệ để phân mảnh và cấp phát CSDL HĐT PT được thực hiện đồng thời. Cách tiếp cận này sử dụng các thông tin về truy vấn tại các trạm cũng như thông tin chi phí kết nối giữa các trạm trong để tạo ra một phương án phân mảnh dọc và cấp phát, đây cũng là một hướng tiếp cận heuristic.

Các phần tiếp theo của bài báo được tổ chức như sau. Phần 2 sẽ giới thiệu về phân mảnh dọc và cấp phát trong CSDL HĐT PT. Phần 3 trình bày các khái niệm cơ bản sẽ được sử dụng trong phân mảnh và cấp phát, trong phần này cũng giới thiệu một ví dụ để sử dụng minh họa cho khái niệm cũng như cho thuật toán trình bày trong phần 5. Phần 4 giới thiệu về mô hình chi phí, trong đó đề cập đến các công thức tính toán chi phí làm cơ sở cho quyết định chọn phương án heuristic. Phần 5 trình bày thuật toán đề xuất và minh họa thuật toán bởi ví dụ. Phần 6 là các kết luận và hướng phát triển.

2. PHÂN MẢNH DỌC VÀ CẤP PHÁT

2.1. Phân mảnh dọc

Dữ liệu trong CSDL HĐT bao gồm một tập các đối tượng được đóng gói, mỗi đối tượng bao gồm các thuộc tính và các phương thức. Các đối tượng được tạo ra từ các lớp, các lớp được phân cấp theo tính chất kế thừa. Một lớp trong quan hệ thứ tự phân cấp được biểu diễn bởi $C = (K, A, M, I)$ trong đó K là tập các định danh, A là tập các thuộc tính, M là tập các phương thức và I là tập các đối tượng được định nghĩa bởi A và M . Phân mảnh dọc của C là $F_v = \{K, A', M', I\}$ trong đó A' là tập con của A , M' là tập con của M .

Các thuộc tính trong một lớp được chia thành hai loại: đơn giản và phức tạp. Thuộc tính đơn giản là thuộc tính có miền giá trị là các kiểu nguyên thủy như int, long, float, double, boolean, char, string ... Thuộc tính phức tạp là thuộc tính có miền giá trị không phải là các kiểu nguyên thủy, mà là tham chiếu tới các đối tượng khác thông qua các định danh của chúng.

Các phương thức trong một lớp cũng được chia thành hai loại: đơn giản và phức tạp. Phương thức đơn giản là phương thức khi thực hiện không gọi các phương thức khác. Phương thức phức tạp là phương thức khi thực hiện gọi phương thức khác trong cùng lớp đó hoặc các phương thức của lớp khác.

Trong bài báo này chúng tôi tập trung vào trường hợp cả thuộc tính và phương thức đều là đơn giản, đây là cơ sở để giải quyết các trường hợp còn lại, khi thuộc tính và phương thức là phức tạp.

2.2. Cấp phát

Cấp phát là định vị và phân bố các mảnh đã được phân chia vào các trạm tương ứng trong mạng liên kết. Giả sử CSDL được phân thành các mảnh $\{F_1, \dots, F_v\}$ và mạng có các trạm S_1, \dots, S_k , việc cấp phát là gán các mảnh F_i vào các trạm S_j sao cho phù hợp về mặt logic và đảm bảo cân bằng tải, nghĩa là xây dựng một ánh xạ từ tập các mảnh vào tập các trạm $\lambda : \{1, \dots, v\} \rightarrow \{1, \dots, k\}$.

Bài toán cấp phát trong CSDL HĐT PT gồm cả cấp phát phương thức và lớp các đối tượng. Bài toán cấp phát phương thức có liên quan chặt chẽ với bài toán cấp phát lớp do vấn đề bao gói. Vì thế cấp phát các lớp kéo theo việc cấp phát phương thức cho các lớp chủ tương ứng của chúng. Nhưng vì các ứng dụng trên các CSDL HĐT có kích hoạt phương thức, việc cấp phát phương thức ảnh hưởng đến hiệu năng các ứng dụng. Các ứng dụng truy cập các thuộc tính và phương thức của một lớp C được chia thành ba loại.

Các ứng dụng thực hiện trực tiếp trên lớp C .

Các ứng dụng thực hiện trên các lớp con của lớp C .

Những ứng dụng thực hiện trên các phương thức của lớp khác trong cơ sở dữ liệu và sử dụng phương thức của lớp C .

Việc cấp phát các phương thức cần truy xuất nhiều lớp nằm ở những vị trí khác nhau là một vấn đề khó. Bài toán cấp phát đã được chứng minh là bài toán NP-đầy đủ [1]. Các hướng tiếp cận cho bài toán cấp phát là nhằm giảm chi phí cấp phát dữ liệu trong các hệ thống phân tán. Chi phí cấp phát là tổng của các chi phí thành phần: chi phí lưu trữ dữ liệu, chi phí xử lý truy vấn, chi phí truyền dữ liệu giữa các trạm.

3. MÔ TẢ MỘT SỐ DỮ LIỆU ĐẦU VÀO

Sự phân mảnh và cấp phát các lớp đối tượng thường được thực hiện dựa vào các mẫu truy vấn thường xuyên của các ứng dụng. Với tính đóng gói của các đối tượng, các ứng dụng thường truy cập các đối tượng thông qua phương thức.

Ví dụ: Giả sử CSDL gồm ba lớp lớp: NhanSu, GiaoVien và CanBo. Hai lớp GiaoVien và CanBo kế thừa từ lớp NhanSu. Trong lớp NhanSu chỉ có các thuộc tính và các phương thức đơn giản. Liệt kê các thuộc tính và các phương thức của các lớp như sau:

NhanSu={{soCMT, ten, ngaysinh, diachi}, {laySoCMT(), layTen(), layNgaysinh(), layDiachi()}}

GiaoVien = {{masoGV, loaiGV, khoa, dsMonday}, {layMasoGV(), layLoaiGV(), layKhoa(), layDSMonday}}

CanBo = {{masoCB, phongban, luong}, {layMasoCB(), layPhongBan(), tinhLuong()}}

Dữ liệu đầu vào cho bài toán được mô tả như sau.

3.1. Sự sử dụng thuộc tính của phương thức

Ma trận sử dụng thuộc tính của các phương thức MAU (Method Attribute Usage) biểu diễn sự sử dụng thuộc tính bởi các phương thức. Trong ma trận MAU, tiêu đề các hàng và các cột là các phương thức và các thuộc tính, giá trị 1 chỉ ra phương thức truy cập thuộc tính tương ứng, ngược lại là 0. Bảng 1 là một ví dụ về ma trận MAU của lớp NhanSu, vì các phương thức sử dụng ở đây là các phương thức cơ bản nên mỗi phương thức chỉ sử dụng 1 thuộc tính.

Bảng 1. Ma trận MAU của lớp NhanSu.

	a ₁ (soCMT)	a ₂ (ten)	a ₃ (ngaysinh)	a ₄ (diachi)
m ₁ (laySoCMT())	1	0	0	0
m ₂ (m.layTen())	0	1	0	0
m ₃ (layNgaysinh())	0	0	1	0
m ₄ (layDiachi())	0	0	0	1

3.2. Sự sử dụng và tần suất truy cập phương thức của truy vấn

Với tính đóng gói của các đối tượng, các ứng dụng chỉ truy cập được các đối tượng thông qua các phương thức. Sự sử dụng phương thức bởi truy vấn được biểu diễn bởi giá trị QMU (Query Method Usage). QMU(q_i, m_j) có giá trị 1 nếu truy vấn q_i có sử dụng phương thức m_j, trường hợp ngược lại QMU(q_i, m_j) có giá trị là 0. Giả sử có 3 truy vấn q₁, q₂, và q₃ tới lớp NhanSu.

q₁: Đưa ra số chứng minh thư (CMT) của tất cả những nhân sự trên trên 55 tuổi.

q₂: Đưa ra số chứng minh nhân dân, tên và địa chỉ của những người ở một vùng nào đó.

q_3 : Liệt kê tên những người sẽ đạt 60 tuổi vào một năm nào đó.

Có 4 truy vấn q_1, q_2, q_3, q_4 đến lớp GiaoVien

q_1 : Đưa ra loại giáo viên (trợ giảng, giảng viên, chuyên gia ...) của một giáo viên có mã số cho trước.

q_2 : Đưa ra danh sách môn dạy của một loại giáo viên.

q_3 : Đưa ra khoa của 1 giáo viên có mã số cho trước.

q_4 : Liệt kê các mã số giáo viên có tuổi lớn hơn hoặc bằng 70.

Có 3 truy vấn q_1, q_2, q_3 đến lớp CanBo

q_1 : Đưa ra số CMT và mã số cán bộ của các cán bộ có tuổi lớn hơn hoặc bằng 50.

q_2 : Liệt kê tên và mã số cán bộ của tất cả các cán bộ trong 1 phòng ban.

q_3 : Liệt kê mã số cán bộ của các cán bộ có lương nhỏ hơn 3 triệu.

Ma trận QMU của các lớp NhanSu là Bảng 2

Bảng 2. Ma trận QMU của lớp NhanSu.

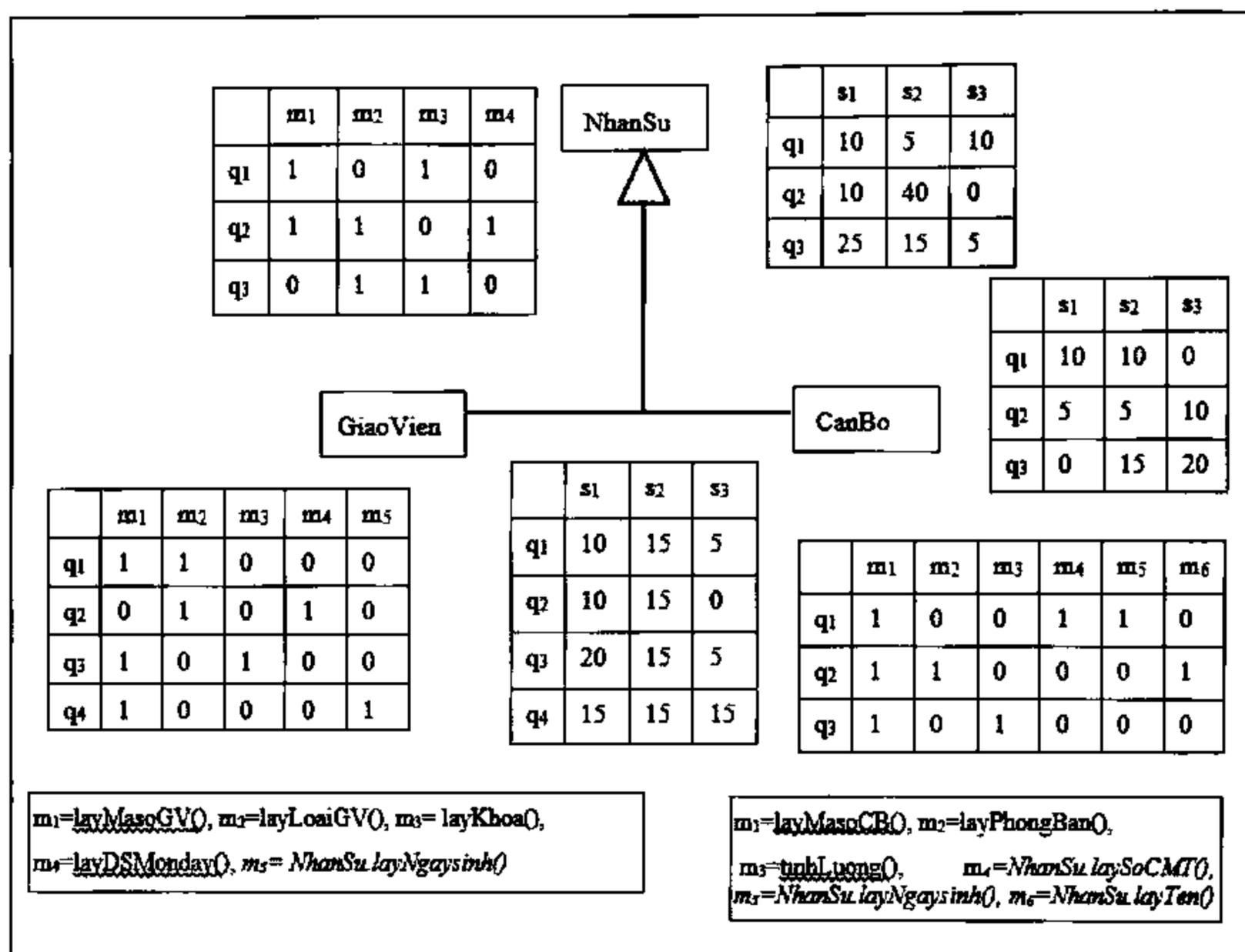
	m_1	m_2	m_3	m_4
q_1	1	0	1	0
q_2	1	1	0	1
q_3	0	1	1	0

Tần suất truy cập vào các trạm của các phương thức được biểu diễn bởi giá trị QSF (Query Site Frequency). $QSF(q_i, s_j)$ là tần suất truy cập vào trạm s_j của truy vấn q_i , ví dụ về ma trận tần suất truy cập trạm được chỉ ra trong Bảng 3.

Bảng 3. Ma trận QSF của lớp NhanSu.

	s_1	s_2	s_3
q_1	10	5	10
q_2	5	40	0
q_3	25	15	5

Ma trận sử dụng phương thức của ứng dụng và ma trận tần suất của các ứng dụng tại các trạm trong ba lớp NhanSu, GiaoVien và CanBo như Hình 1. Chú ý rằng các truy vấn vào 2 lớp GiaoVien và CanBo sẽ sử dụng một số phương thức từ lớp cha NhanSu, vì vậy ma trận QMU của lớp GiáoVien và CanBo sẽ phải bổ sung thêm các phương thức này. Ma trận QMU của lớp GiaoVien có 5 cột, 4 cột đầu biểu diễn sự truy cập các phương thức m_1, m_2, m_3, m_4 của chính lớp Giáo Viên, cột cuối cùng sau biểu diễn sự truy cập phương thức m_5 của lớp NhanSu. Tương tự, ma trận QMU của lớp CanBo có 6 cột trong đó 3 cột sau là biểu diễn sự truy cập các phương thức của lớp NhanSu. Trong Hình 1, các phương thức gọi từ lớp cha NhanSu được in nghiêng. Cũng chú ý rằng các chỉ số của các phương thức và truy vấn là của riêng từng lớp (m_i của NhanSu là khác m_j của GiaoVien, q_i của NhanSu là khác q_j của GiaoVien).



Hình 1. Ma trận QMU và QSF của các lớp.

3.4 Chi phí giao tiếp của các trạm

Ma trận chi phí giao tiếp giữa các trạm kí hiệu là SSC (Site Site Cost) biểu diễn chi phí giao tiếp giữa các trạm. Ví dụ về một ma trận chi phí giữa các trạm như Bảng 4.

Bảng 4. Ma trận SSC.

	s1	s2	s3
s1	0	50	70
s2	50	0	30
s3	70	30	0

4. MÔ HÌNH CHI PHÍ

Chi phí đáng kể nhất trong CSDL HĐT PT là chi phí truyền dữ liệu giữa các trạm. Ở đây, chúng tôi xác định hàm chi phí để tính tổng các chi phí gọi các phương thức tại các trạm từ xa. Chi phí gọi một phương thức được biểu diễn bằng tổng chi phí giao tiếp giữa các trạm chứa những phương thức được gọi (sử dụng) và trạm thực hiện việc gọi chúng. Tiếp đến khi tính thời gian thực hiện của bất kỳ phương thức nào, kiểu của nó sẽ được xác định trước tiên. Nếu nó là

một phương thức đơn giản, chi phí được xem xét chỉ là chi phí truyền dữ liệu kết quả từ việc thực hiện phương thức tới trạm đang gọi. Nếu nó là phương thức phức tạp, ta phải thêm vào chi phí thực hiện phương thức được gọi. Cần chú ý thêm về yêu cầu truyền các đối tượng từ xa, điều này xảy ra khi một phương thức yêu cầu các đối tượng trên các trạm từ xa để thực hiện, trong trường hợp này phải thêm vào chi phí truyền các đối tượng. Tóm lại, chi phí của bất kì phương thức nào cũng bao gồm:

- Chi phí dữ liệu trả về trạm đang gọi
- Chi phí truyền các đối tượng từ xa
- Chi phí của các phương thức được gọi

Trong CSDL có nhiều lớp, để phân biệt các tham số trong từng lớp, cần đưa thêm chỉ số của lớp, ví dụ m_j^i để chỉ phương thức m_j thuộc lớp thứ i . Các kí hiệu sử dụng được mô tả trong Bảng 5.

Bảng 5. Các kí hiệu sử dụng.

Kí hiệu	Mô tả ý nghĩa
C_i	Lớp thứ i trong CSDL HĐT PT.
M^i	Số lượng các phương thức của lớp C_i .
m_j^i	Phương thức thứ j của lớp C_i .
Q^i	Số lượng các truy vấn của lớp C_i .
q_j^i	Truy vấn thứ j của lớp C_i .
S	Số lượng các trạm
s_k	Trạm thứ k
MAU^i	Là ma trận biểu diễn sự sử dụng thuộc tính của phương thức của lớp C_i .
QMU^i	Là ma trận biểu diễn sự sử dụng phương thức của truy vấn của lớp C_i .
QSR^i	Là ma trận biểu diễn tần suất truy cập vào trạm của các truy vấn của lớp C_i .
SSC	Là ma trận chi phí giao tiếp giữa các trạm

Chi phí truy cập một phương thức m_j^i ở một trạm s_k là tổng tần suất truy cập của các truy vấn có sử dụng phương thức m_j^i trên trạm s_l , chi phí này được thiết lập như sau:

$$request^i(s_k, m_j^i) = \sum_{l=1}^{Q^i} QSF^i(q_l, s_k) * QMU^i(q_l, m_j^i)$$

Để thiết lập chi phí truy cập mỗi phương thức của một lớp C_i từ các trạm chúng ta sẽ xây dựng ma trận $request^i$, ma trận này chính là tích của 2 ma trận QSF^i (là ma trận chuyển vị của ma trận QSR^i) và ma trận QMU^i .

Chi phí khi định vị một phương thức m_j^i vào một trạm s_k là chi phí truy cập vào phương thức m_j^i từ tất cả các trạm $s_l \neq s_k$, chi phí này được thiết lập như sau:

$$\text{pay}^i(s_k, m_j^i) = \sum_{l=1, l \neq k}^S \text{request}^i(s_k, m_j^i) * \text{SSC}(s_k, s_l)$$

Để thiết lập chi phí định vị mỗi phương thức của một lớp C_i vào các trạm chúng ta sẽ xây dựng ma trận pay^i , ma trận này chính là tích của 2 ma trận request^i và SSC .

Dựa vào ma trận pay^i để xác định phương án cấp phát các phương thức của một C_i . Thuật toán đề nghị là một thuật toán heuristic, mục tiêu của thuật toán là định vị m_j^i vào trạm s_k mà giá trị $\text{pay}^i(s_k, m_j^i)$ là bé nhất.

5. THUẬT TOÁN PHÂN MẢNH VÀ CẤP PHÁT ĐỒNG THỜI

5.1. Xây dựng thuật toán

Thuật toán đề xuất theo hướng tiếp cận heuristic như sau: Dựa vào ma trận pay^i , định vị phương thức m_j^i vào trạm s_k mà chi phí giao tiếp là bé nhất. Thiết lập một phương án định vị, sau đó gom cụm các phương thức ở cùng một trạm vào một mảnh. Trong mỗi mảnh, với từng phương thức xác định các thuộc tính mà các phương thức này sử dụng để đưa vào mảnh này. Thuật toán phân mảnh và cấp phát đồng thời được xây dựng như sau.

Input: Lớp C_1 cần phân mảnh trong CSDL các lớp, các ma trận MAU, QMU, QSF của các lớp, ma trận SSC.

Output: Phân mảnh dọc và cấp phát cho lớp C_1 .

Các bước thực hiện:

// Bước 1: Thay đổi ma trận sử dụng phương thức và ma trận tần

//suất truy cập phương thức của lớp C_1 xét đến các truy vấn từ

// các lớp con cháu của C_1

for each C_{11} là lớp con cháu của C_1 **do**

for each q_k^u truy vấn vào lớp C_{11} **do**

if q_k^u sử dụng các phương thức m_j^i của lớp C_i

begin

 Thêm 1 dòng tương ứng q_k^u vào QMU¹

 Thêm 1 dòng tương ứng q_k^u vào QSF¹

end {if q_k^u }

```
        end {for  $q_k^u$ }
    end {for  $C_{11}$ }
//Bước 2: Xây dựng ma trận  $request^1$  của lớp  $C_1$  bằng cách nhân  $\Delta$ 
//ma trận  $QMU^1$  và  $QSF^1$ 
 $request^1 = Nhan2matran(QMU^1, QSF^1)$  //Phép nhân 2 ma trận 2 chiều
//Bước 3: Xây dựng ma trận  $pay^1$  của lớp  $C_1$  bằng cách nhân 2 ma
//trận  $request^1$  và  $SSC$ 
 $pay^1 = Nhan2matran(request^1, SSC)$  //Phép nhân 2 ma trận 2 chiều
//Bước 4: Xác định phương án cấp phát dựa vào ma trận  $pay^1$ 
//Tìm giá trị nhỏ nhất của mỗi cột trong ma trận  $pay^1$ 
for each  $m_j^i$  do
    begin
        Chọn  $s_k$  mà  $pay^1(s_k, m_j^i)$  có giá trị bé nhất
        Cấp phát  $m_j^i$  vào trạm  $s_k$ 
        Thêm  $m_j^i$  vào  $F_k$ 
    end {for  $m_j^i$ }
Thêm vào mỗi mảnh phương thức truy cập định danh
//Bước 5: Dựa vào ma trận MAU thêm các thuộc tính vào các mảnh
for each  $m_j^i$  do
    for each  $a_i^1$  do
        if  $MAU(m_j^i, a_i^1) = 1$ 
            begin
                Thêm  $a_i^1$  vào mảnh có  $m_j^i$ 
                Định vị  $a_i^1$  vào trạm mà  $m_j^i$  định vị
            end {if}
        end {for  $a_i^1$ }
    end {for  $m_j^i$ }
end {thuật toán}
```

5.2. Ví dụ minh họa

Trong ví dụ này chỉ đưa ra phân mảnh của lớp NhanSu nên tạm thời bỏ 1 chỉ số phía trên của các tham số.

Khi xét các truy vấn đến lớp Giáo viên, có truy vấn q_4 truy cập đến phương thức m_4 của lớp NhanSu, như vậy sẽ bổ sung vào ma trận QMU một dòng. Tương tự như vậy, khi xét đến lớp CanBo sẽ bổ sung thêm 2 dòng tương ứng với q_5 và q_6 (chính là q_1 và q_2 của lớp CanBo). Ma trận sử dụng phương thức QMU của lớp NhanSu trong Bảng 2 sẽ được điều chỉnh như Bảng 6. Tương ứng ma trận QSF cũng tương ứng thêm 3 dòng cho các truy vấn vừa xét đến. Ma trận SQF (ma trận chuyển vị của ma trận tần suất truy cập phương thức QSF) của lớp NhanSu sẽ được điều chỉnh như Bảng 7.

Bảng 6. Ma trận QMU thay đổi sau khi xét quan hệ kế thừa.

	m_1	m_2	m_3	m_4
q_1	1	0	1	0
q_2	1	1	0	1
q_3	0	1	1	0
q_4	0	0	1	0
q_5	1	0	1	0
q_6	0	1	0	0

Bảng 7. Ma trận SQF thay đổi sau khi xét quan hệ kế thừa (ma trận chuyển vị của ma trận QSF ở Bảng 3)

	q_1	q_2	q_3	q_4	q_5	q_6
s_1	10	10	25	15	10	5
s_2	5	40	15	15	10	5
s_3	10	0	5	15	0	10

Kết hợp với ma trận SQF được QMU để xây dựng ma trận request như Bảng 8.

Bảng 8. Ma trận request.

	m_1	m_2	m_3	m_4
s_1	35	55	50	20
s_2	30	30	27	15
s_3	5	5	5	0

Kết hợp ma trận SSC trong Bảng 4 được ma trận pay như Bảng 9.

Phương án cấp phát như sau: m_1 được chọn định vị ở s_3 , m_2 chọn định vị ở s_2 , m_3 chọn định vị ở s_2 , m_4 chọn định vị ở s_3 . Như vậy m_1 và m_4 được ghép vào cùng một mảnh, m_2 và m_3 được ghép vào mảnh thứ 2. Vì m_1 là phương thức truy cập vào định danh nên m_1 được bổ sung

vào mảnh của m_2 và m_3 . Vậy các phương thức được phân mảnh dọc thành 2 mảnh $F_1 = \{m_1, m_4\}$, $F_2 = \{m_1, m_2, m_3\}$. Căn cứ vào ma trận MAU trong bảng 1 sẽ tiếp tục phân a_1, a_4 vào F_1 , phân a_1, a_2, a_3 vào F_2 . Kết quả là $F_1 = \{a_1, a_4, m_1, m_4\}$, $F_2 = \{a_1, a_2, a_3, m_1, m_2, m_3\}$.

Bảng 9. Ma trận pay.

	m_1	m_2	m_3	m_4
s_1	7950	9750	11850	4000
s_2	5800	6750	6900	3500
s_3	5750	6900	7950	3100

5.3. Đánh giá thuật toán

Mục tiêu của thuật toán là tối thiểu hóa chi phí, độ phức tạp của thuật toán chính là độ phức tạp của phép nhân hai ma trận. Nếu lớp cần phân mảnh có M phương thức, số lượng truy vấn là Q và số các trạm là S thì độ phức tạp của thuật toán là $M*Q*S$. Số lượng các mảnh theo thuật toán này tối đa chỉ bằng số trạm.

6. KẾT LUẬN

Hầu hết các giải thuật từ trước đến nay chia giai đoạn phân mảnh và cấp phát thành hai giai đoạn độc lập, phân mảnh hoàn thành rồi mới đến cấp phát. Trong giai đoạn phân mảnh không tính đến chi phí về giao tiếp giữa các trạm, chi phí này chỉ được xác định khi thực hiện cấp phát. Tuy nhiên việc cấp phát chi tối ưu khi phân mảnh hợp lí có tính đến chi phí giao tiếp giữa các trạm để đạt được chi phí bé nhất. Bài báo đề cập đến thuật toán heuristic trong đó việc phân mảnh vừa cấp phát thực hiện đồng thời giúp cho việc phân mảnh các đối tượng hiệu quả hơn.

Hướng phát triển của các tác giả là tiếp tục xét các trường hợp còn lại của cấu trúc lớp, lớp có thuộc tính phức tạp và phương thức đơn giản, lớp có thuộc tính đơn giản và phương thức phức tạp, lớp có thuộc tính phức tạp và phương thức phức tạp.

TÀI LIỆU THAM KHẢO

1. Ozsu M. T. and Valduriez P. - Principles of Distributed Database Systems, Third Edition, Springer, New York, 2011, pp. 845.
2. Navathe S. B. and Ra M. - Vertical partitioning for database design: Agraphical algorithm, SIGMOD Record 18 (2) (1989) 440-450.
3. Hoffer H. A. and Severance D. G. - The Use of Cluster Analysis in Physical Database Design, Proc. First Int. Conf. on Very Large Database, ACM New York, 1975, pp. 69-86.
4. Karlapalem K. and Li Q. - Partitioning Schemes for Object Oriented Database, 5th International Workshop on Research Issues on Data Engineering: Distributed Object Management, IEEE, 1995, pp. 42-49.
5. Ezeife C.I., Ken Barker - Distributed Object Based Design: Vertical Fragmentation of Classes, Distributed and Parallel Databases 6 (4) (1998) 317-350.
6. Lee Soonmi and Lim Haechull - Attribute Partitioning Algorithm in DOODB, In Proc. First Int. Conf. on Distributed and Parallel Databases, IEEE, 1997, pp. 702-707.

7. Rajan John and Dr. V. Saravanan - Vertical Partitioning in Object Oriented Databases Using Intelligent Agents, IJCSNS International Journal of Computer Science and Network Security 8 (10) (2008) 205-210.
8. Lee Soonmi and Lim Haechull - Allocation of Classes in distributed object-oriented databses, 10th ACIS Internatonal Conference on software Enginneering, IEEE, 2009, pp. 237-242.
9. Sarhan A. - A New Allocation Technique for Methods and Attributes in Distributed Object-Oriented Database using Genetic Algorithms, The International Arab Journal of Information Technology 6 (1) (2009) 17-26.
10. Barkèr K. and Bhar S. - Agraphical approach to allocation class fragments in distributed object-oriented base systems, Distributed and Parallel Databases 10 (3) (2001) 207-239.
11. Hui Ma, Klaus-Dieter Schewe and Markus Kirchberg - A Heuristic Approach to Vertical Fragmentation Incorporating Query Information, Massey University, New Zealand, 2006, pp. 69-76.

ABSTRACT

FRAGMENTATION AND ALLOCATION ALGORITHM IN THE DISTRIBUTED OBJECT-ORIENTED DATABASE: CASE OF SIMPLE ATTRIBUTES AND SIMPLE METHODS

Mai Thúy Nga^{1,*}, Đoàn Văn Ban², Nguyễn Mạnh Hùng³

¹*Information Technology Department, ThangLong University, Nghiem Xuan Yem Road, Hoang Mai District, Hà Nội*

²*Institute of Information Technology, 18 Hoang Quoc Viet Road, Cau Giay District, Hà nội*

³*Le Quy Don Technical University, 236 Hoang Quoc Viet Road, Cau Giay District, Hà nói*

*Email: mai_nga@yahoo.com

Nowadays, information systems are often built based on the object-oriented architecture and distributed in the network, they consist of many diverse nonhomogeneous computational basis. In such environments, the systems with distributed object-oriented database provide a reliable and effective mechanism to organize data storage and handle inquires to very large-volume of different objects. With the fundamental characteristics of object orientation such as encapsulation, inheritance, class hierarchy, the design aspect in the distributed object-oriented database appears more complicated matters than the normal design approach in the relationship database. The design in distributed object-oriented database has two phases: Class Fragmentation and Class Allocation. Class Fragmentation and Allocation aim to improve the system performance. The two phases are normally executed in the sequence mode, only when the fragmentation is complete the allocation starts. However, this paper presents an algorithm of which the fragmentation and allocation phases are executed simultaneously in the distributed object-oriented database, in case simple attributes and simple methods.

Keywords: object-oriented database, distributed database, distributed object-oriented database, fragmentation, allocation.