Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.00067

PHÂN MẢNH VÀ ĐỊNH VỊ DỮ LIỆU PHÂN TÁN BẰNG TÁC TỬ DI ĐỘNG

Trần Đình Toàn[1], Nguyễn Mậu Hân[1]
[1] Đại học Khoa học Huế
toan.tranbl@gmail.com, nmhan2009@gmail.com

TÓM TẮT *— Trong bài báo này, chúng tôi giới thiệu giải pháp phân mảnh dữ liệu và định vị dữ liệu sử dụng tác tử di động trong hệ CSDL phân tán, việc sử dụng tác tử di động nhằm làm tăng tính chủ động, linh hoạt và khả năng phản ứng lại sự thay đổi của hệ thống. Chúng tôi tập trung vào phân mảnh và cấp phát dữ liệu với mục đích cấp phát dữ liệu tối ưu, từ đó, nhằm làm giảm chi phí truyền dữ liệu, nâng cao hiệu suất của hệ thống.*

Từ khóa *— Distributed database, fragmentation, allocation, mobile agents, optimization.*

I. GIỚI THIỆU

Trong những năm gần đây, đã có những tiến bộ đáng kể trong sự phát triển của các hệ thống mạng máy tính. Cùng với xu thế toàn cầu hóa trong mọi lĩnh vực, đặc biệt là về thương mại, cơ sở dữ liệu (CSDL) phân tán đã trở thành một lĩnh vực thu hút nhiều sự quan tâm của các nhà nghiên cứu lý thuyết lẫn các nhà sản xuất phần mềm. Công nghệ về các hệ CSDL phân tán là sự hợp nhất của hai hướng tiếp cận đối với quá trình xử lý dữ liệu: công nghệ CSDL và công nghệ mạng máy tính [2]. Chúng ta có thể hiểu một hệ CSDL phân tán là một CSDL có thể được lưu trữ trong nhiều máy tính đặt tại các vị trí vật lý khác nhau, hoặc có thể được phân tán qua một mạng các máy tính kết nối với nhau. Như vậy, một hệ CSDL phân tán không phải là một hệ thống mà trong đó CSDL lại chỉ nằm ở một nút của mạng. Trong một số trường hợp, người ta cho rằng CSDL cần được nhân bản để lưu trữ tại các nút khác trong mạng sẽ tốt hơn là phải tải khối lượng dữ liệu đó từ vị trí trung tâm đến các vị trí khác. Tuy nhiên, nếu cần thiết phải cập nhật dữ liệu thì khi nhân bản quá nhiều đòi hỏi phải cài đặt các phương thức điều khiển đồng thời và ủy thác hợp lý. Vì thế vấn đề khó khăn trong việc thiết kế CSDL phân tán là giải bài toán cấp phát dữ liệu trên các vị trí (site) sao cho hợp lý. Bài toán này thuộc loại NP-đầy đủ, vì vậy các giải pháp được đề xuất đều dựa trên các thuật giải heuristic.

II. CÁC CÔNG TRÌNH LIÊN QUAN

Kỹ thuật phân mảnh để phân vùng các quan hệ của CSDL phân tán ở giai đoạn đầu khi chưa có các thống kê truy cập dữ liệu và tần số thực hiện truy vấn. Sử dụng kỹ thuật này nhằm đồng bộ giữa phân mảnh các quan hệ và cấp phát các mảnh vào các vị trí của CSDL phân tán [3].

Cấp phát dữ liệu là để xác định việc chuyển các mảnh tại các vị trí khác nhau để giảm thiểu tổng chi phí truyền dữ liệu có liên quan trong việc thực hiện một tập các câu truy vấn [4].

Các tác tử di động có thể gửi tới host đích dữ liệu mang theo để tính toán tại các vị trí ở xa. Tác tử di động chuyển dữ liệu tới CSDL phân tán từ xa, không phải là CSDL tới dữ liệu. Do đó, hệ thống tiết kiệm băng thông và khắc phục độ trễ mạng [5]. Tầm quan trọng của các hệ thống CSDL phân tán tăng thêm với sự phát triển của công nghệ mạng. Sự cấp phát hiệu quả các mảnh dữ liệu đóng vai trò quan trọng trong hoạt động của CSDL về hiệu suất và chi phí. Sự phân mảnh dữ liệu và cấp phát mảnh với chi phí tối thiểu cho cả dữ liệu có cấu trúc và không có cấu trúc [6]. Sử dụng hệ thống đa tác tử cho việc quản lý CSDL phân tán làm giảm lưu lượng băng thông mạng và giảm truyền dữ liệu. Công nghệ đa tác tử là một phương pháp thay thế cho các hệ thống truyền thống client-server [7]. Sử dụng các kỹ thuật phân nhóm hướng tri thức cho vấn đề phân mảnh trong hệ thống CSDL phân tán [8].

III. PHÂN MẢNH

Trong hệ CSDL phân tán hỗ trợ phân mảnh nếu một quan hệ có thể được phân thành các phần gọi là các mảnh cho mục đích lưu trữ vật lý nhằm làm giảm không gian lưu trữ. Các mảnh có thể được lưu trữ ở các site khác nhau nơi có thể có nhiều truy xuất thường xuyên hơn do lưu lượng mạng thấp và tăng hiệu suất hệ thống. Phân mảnh dữ liệu có thể thực hiện theo chiều ngang hay theo chiều dọc.

Việc phân mảnh các quan hệ sẽ cho phép thực hiện song song một tập câu vấn tin bằng cách chia nó ra thành một tập các câu vấn tin con hoạt động trên các mảnh. Vì thế việc phân mảnh sẽ làm tăng mức độ hoạt động đồng thời của hệ thống.

Các công trình đã công bố chủ yếu tập trung vào phân mảnh ngang. [2] Trong khi đó có hai chiếc lược phân mảnh là phân mảnh ngang và phân mảnh dọc, trong một số trường hợp có thể dùng hỗn hợp cả hai chiếc lược này, trong đó

Phân mảnh ngang: có hai chiến lược là phân mảnh ngang nguyên thủy và phân mảnh ngang dẫn xuất.

Phân mảnh dọc: Phân mảnh dọc phức tạp hơn phân mảnh ngang là do tổng số chọn lựa có thể có của phân hoạch rất lớn. Vì vậy để có được lời giải tối ưu cho bài toán phân mảnh dọc thật sự là rất khó. Do đó phải dùng các phương pháp heuristic. Chúng ta đưa ra hai loại heuristic cho phân mảnh dọc các quan hệ toàn cục là nhóm thuộc tính và tách mảnh.

Ví dụ 1: Cho lược đồ CSDL

EMP(ENO, ENAME, TITLE)

ASG(ENO, PNO, RESP, DUR)

PROJ(PNO, PNAME, BUGGET, LOC)

PAY(TITLE, SAL)

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elec. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Ch	Elect. Eng.
E7	R. David	Mech. Eng
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

TITLE	SAL
Elec. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Ví dụ 2: Phân mảnh ngang

Quan hệ EMP được phân thành 4 mảnh

EMP_1 chứa các nhân viên có TITLE= “Elect. Eng.”

EMP_2 chứa các nhân viên có TITLE= “Syst. Anal.”

EMP_3 chứa các nhân viên có TITLE= “Mech. Eng.”

EMP_4 chứa các nhân viên có TITLE= "Programmer"

Phân mảnh ngang được định nghĩa như là phép chọn $\delta_F(R)$ trong đại số quan hệ

$EMP_1 = \delta_{TITLE= \text{"Elect. Eng."}}(EMP)$

$EMP_2 = \delta_{TITLE= \text{"Syst. Anal."}}(EMP)$

$EMP_3 = \delta_{TITLE= \text{"Mech. Eng."}}(EMP)$

$EMP_4 = \delta_{TITLE= \text{"Programmer"}}(EMP)$

EMP_1

ENO	ENAME	TITLE
E1	J. Doe	Elec. Eng.
E6	L. Ch	Elect. Eng.

EMP_2

ENO	ENAME	TITLE
E2	M. Smith	Syst. Anal.
E5	B. Casey	Syst. Anal.
E8	J. Jones	Syst. Anal.

EMP_3

ENO	ENAME	TITLE
E3	A. Lee	Mech. Eng.
E7	R. David	Mech. Eng

EMP_4

ENO	ENAME	TITLE
E4	J. Miller	Programmer

Ví dụ 3: Phân mảnh dọc

Quan hệ PROG được chia thành 2 mảnh dọc

$PROJ_1$: chứa thông tin về ngân sách (BUDGET) các dự án.

$PROJ_2$: chứa thông tin về tên (PNAME) và vị trí (LOC) các dự án

Phân mảnh dọc được định nghĩa như là phép chiếu $\Pi_A(R)$ trong đại số quan hệ

$PROJ_1 = \Pi_{PNO, BUDGET}(PROJ)$

$PROJ_1 = \Pi_{PNO, PNAME, LOC}(PROJ)$

$PROJ_1$ $PROJ_2$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris

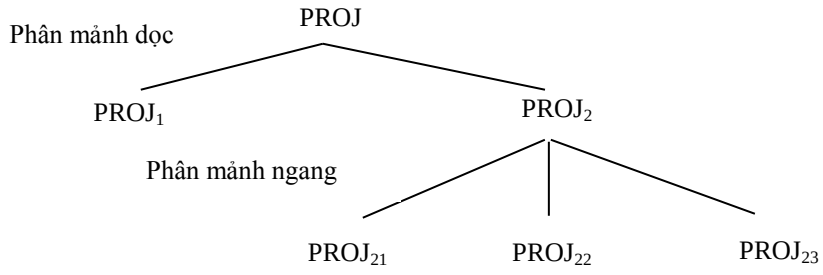
Ví dụ 4: Phân mảnh hỗn hợp

Quan hệ $PROJ_2$ được chia thành 3 mảnh ngang

$PROJ_{21}$: chứa thông tin về mã, tên của các dự án ở (LOC) "Montreal"

$PROJ_{22}$: chứa thông tin về mã, tên của các dự án ở (LOC) "New York"

$PROJ_{23}$: chứa thông tin về mã, tên của các dự án ở (LOC) "Paris"



$PROJ = PROJ_{21} \cup (PROJ_{21} \cap PROJ_{22} \cap PROJ_{23})$

Phân mảnh hỗn hợp được định nghĩa gồm phép chọn và phép chiếu $\Pi_A(R)$ trong đại số quan hệ $\delta_F(\Pi_{A1, \ldots, An}(R))$ hoặc $\Pi_{A1, \ldots, An}(\delta_F(R))$

$PROJ_{21} = \delta_{LOC= \text{"Montreal"}}(\Pi_{PNO, PNAME, LOC}(PROJ))$

$PROJ_{22} = \delta_{LOC= \text{"New York"}}(\Pi_{PNO, PNAME, LOC}(PROJ))$

$PROJ_{23} = \delta_{LOC= \text{"Paris"}}(\Pi_{PNO, PNAME, LOC}(PROJ))$

Khi phân mảnh ba quy tắc phải tuân thủ để đảm bảo CSDL sẽ không có thay đổi về ngữ nghĩa [2].

Tính đầy đủ (completeness): Nếu một quan hệ R được phân rã thành các mảnh $R_1, R_2, \ldots, R_n$, thì mỗi mục có thể gặp trong R cũng có thể gặp trong một hoặc nhiều mảnh R_i.

Tính tái thiết (reconstruction): Nếu một quan hệ R được phân rã thành các mảnh $R_1, R_2, \ldots, R_n$, thì cần phải định nghĩa một toán tử quan hệ sao cho có thể thiết lập lại quan hệ R từ các mảnh R_i.

Tính tách biệt (disjointness): Nếu quan hệ R được phân rã ngang thành các mảnh $R_1, R_2, \ldots, R_n$, và mục dữ liệu t_i nằm trong mảnh R_j, thì nó sẽ không nằm trong mảnh R_k khác ($k \neq j$). Tiêu chuẩn này đảm bảo các mảnh ngang sẽ tách biệt (rời nhau). Nếu quan hệ R được phân rã dọc, các thuộc tính khoá chính phải được lặp lại trong mỗi mảnh. Vì thế trong trường hợp phân mảnh dọc, tính tách biệt chỉ được định nghĩa trên các trường không phải là khoá chính của một quan hệ.

IV. CẤP PHÁT DỮ LIỆU SỬ DỤNG TÁC TỬ DI ĐỘNG

Bài toán cấp phát

Cấp phát tài nguyên CSDL phân tán cho các nút của một mạng máy tính là một bài toán được nhiều người quan tâm và nghiên cứu rộng rãi. Đây là bài toán cấp phát mảnh, giả sử cho một tập các mảnh $F=\{F_1, F_2, \ldots, F_k\}$ và trong một mạng máy tính bao gồm các vị trí $S=\{S_1, S_2, \ldots, S_n\}$ trên đó có một tập ứng dụng $Q=\{q_1, q_2, \ldots, q_m\}$ đang chạy. Bài toán đặt ra là tìm một phân phối tối ưu của F cho S.

Quá trình phân mảnh gắn liền với quá trình cấp phát và các bài toán cụ thể, cần lưu ý một điều quan trọng là cho đến hiện tại chúng ta vẫn chưa có một thuật toán tổng quát tối ưu cho bài toán phân mảnh tổng quát và cấp phát dữ liệu trên mạng.

Tính tối ưu có thể được định nghĩa với hai giá trị:

Chi phí nhỏ nhất: Hàm chi phí gồm chi phí lưu mảnh F_i tại vị trí S_j, chi phí vấn tin F_i tại vị trí S_j, chi phí cập nhật F_i tại tất cả các vị trí có chứa nó, và chi phí truyền dữ liệu. Vì vậy bài toán cấp phát ở đây là cố gắng tìm một lược đồ cấp phát với hàm chi phí thấp nhất.

Hiệu quả: Chiến lược cấp phát được thiết kế nhằm duy trì hiệu quả là giảm thấp thời gian đáp ứng và tăng tối đa lưu lượng hệ thống tại mỗi vị trí.

Mô hình cấp phát: có mục tiêu là giảm thiểu tổng chi phí xử lý và lưu trữ. Min(Total_Cost) ứng với ràng buộc thời gian đáp ứng, ràng buộc lưu trữ và ràng buộc xử lý. Ma trận x(n, m) thể hiện cách cấp phát mảnh vào các site:

$$x_{ij} = \begin{cases} 1 \text{ nếu mảnh } F_i \text{ được cấp phát tại site } S_j \\ 0, \text{ trong trường hợp ngược lại} \end{cases}$$

Ví dụ 5: Ma trận x(nxm) cho biết mảnh F_i được cấp phát tại site S_j

$$\begin{array}{c} \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Ví dụ 6: Ma trận RM(qxf) dòng là các truy vấn, cột là các mảnh, ma trận thể hiện số lần truy vấn chỉ đọc, phần tử r_{ij} trong ma trận RM thể hiện số lần câu truy vấn q_i thực hiện chỉ đọc tại mảnh F_j.

$$\begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{array} \begin{array}{c} \begin{array}{ccccc} F_1 & F_2 & F_3 & F_4 & F_5 \end{array} \\ \begin{pmatrix} 2 & 0 & 1 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix} \end{array}$$

Ví dụ 7: Ma trận UM(qxn) dòng là các truy vấn, cột là các mảnh, ma trận thể hiện số lần truy vấn cập nhật. Phần tử u_{ij} trong ma trận UM thể hiện số lần câu truy vấn q_i thực hiện cập nhật tại mảnh F_j.

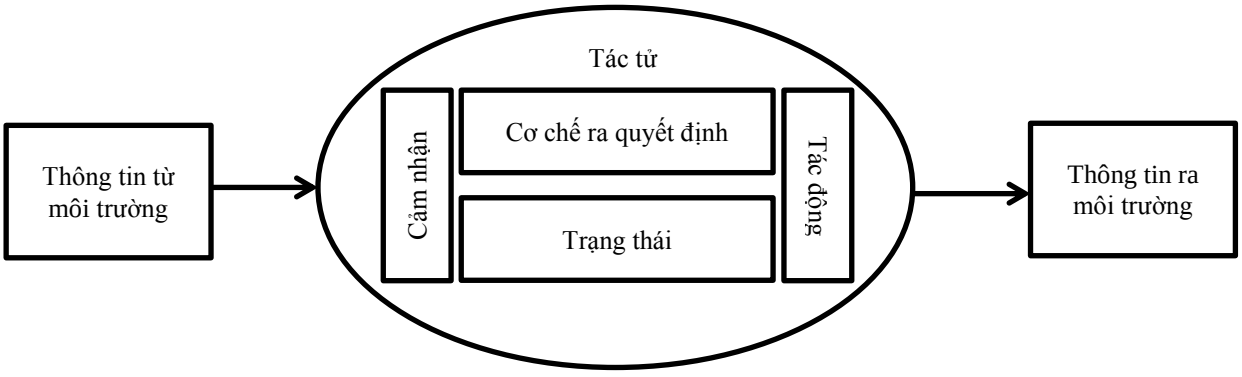
$$\begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{array} \begin{array}{c} \begin{array}{ccccc} F_1 & F_2 & F_3 & F_4 & F_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 2 \\ 0 & 3 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix} \end{array}$$

Hệ thống phân tán có thể được biểu diễn như một đồ thị G=(V, E), trong đó V là tập đỉnh của đồ thị đại diện cho các site, E là tập cạnh của đồ thị đại diện là đường kết nối giữa các đỉnh (sites) của đồ thị. Mỗi cạnh được gán 1 giá trị gọi là chi phí (cost).

Tác tử di động [1]

Tác tử di động là một tác tử có khả năng di chuyển một cách tự trị từ nút mạng này sang nút mạng khác và thực hiện các các công việc được giao thay thế cho con người. Khi di chuyển, các tác tử di động đóng gói mã nguồn, dữ liệu và cả trạng thái thi hành, nhờ vậy tác tử di động có thể dừng việc thi hành đang thực hiện tại máy này, di chuyển sang máy khác và khôi phục lại sự thi hành.

Các tác tử di động có thể giúp làm giảm lưu lượng trên mạng thay vì phải truyền một lượng lớn thông tin trên mạng thì chúng ta chỉ cần truyền chức năng của chúng. Các tác tử di động cũng có thể xử lý trước dữ liệu và chỉ truyền các kết quả. Còn được gọi là nén ngữ nghĩa.



Hình 1. Cấu trúc của tác tử

Từ Hình 1, chúng ta có thể thấy tác tử nhận thông tin từ môi trường (bao gồm thông tin từ các tác tử khác) thông qua cơ quan cảm nhận. Nhờ có cơ chế ra quyết định, tác tử lựa chọn hành động cần thực hiện. Quá trình ra quyết định có thể sử dụng thông tin về trạng thái bên trong của tác tử. Trong trường hợp đó, tác tử lưu trữ trạng thái dưới dạng những cấu trúc dữ liệu riêng. Hành động do cơ chế ra quyết định lựa chọn sau đó được tác tử thực hiện thông qua cơ quan tác động. Cơ chế suy diễn có thể thay đổi cho từng kiểu kiến trúc cụ thể và ảnh hưởng tới những thành phần khác.

Chẳng hạn có thể có kiến trúc trong đó quá trình suy diễn không sử dụng tới trạng thái bên trong và do vậy tác tử không cần lưu giữ các thông tin này. Đối với các tác tử có thêm khả năng khác như học tự động,… kiến trúc tác tử có thể có thêm thành phần riêng để thực hiện các chức năng này.

Tác tử di động là một phần mềm thông minh, nó bao gồm cả các yêu cầu người dùng và kết hợp với tính ưu việt của tác tử. Vì thế truy vấn dữ liệu dựa trên tác tử di động không chỉ đáp ứng các yêu cầu người dùng mà còn khắc phục được các khuyết điểm của các hệ thống thông thường khác.

Chúng ta có thể phân biệt một tác tử từ một hệ chuyên gia bởi thực tế một hệ chuyên gia đóng vai trò như một nhà tư vấn và đưa ra lời khuyên trong việc lựa chọn các giải pháp trong khi một tác tử thông minh có thể hành động để thay đổi môi trường.

Giải quyết bài toán phân tán là đặc trưng hợp tác làm việc của các tác tử di động khi các vấn đề cần sự giải quyết tập thể, cũng như phân tán nguồn tài nguyên sẵn có như tri thức, khả năng và chuyên môn giữa các tác tử.

Vấn đề cấp phát dữ liệu có thể mô hình hóa bằng phương pháp tác tử di động bởi vì nó được mô tả trong thuật toán tìm kiếm cho các tác tử, chúng ta có thể phân bài toán này như bài toán tìm đường đi. Bài toán tìm đường đi bao gồm các thành phần:

Tập N các nút, mỗi nút đại diện cho một trạng thái.

Tập L là tập các liên kết có hướng, mỗi cái đại diện cho một toán tử có sẵn để các tác tử giải quyết vấn đề.

Giả sử chúng ta biết trạng thái ban đầu.

Tập G các nút, mỗi nút đại diện cho một trạng thái mục tiêu.

Với mỗi liên kết, trong số của liên kết được định nghĩa đại diện cho các chi phí áp dụng các toán tử. Chúng ta gọi trọng số của đường liên kết giữa hai nút là khoảng cách giữa hai nút, gọi các nút có hướng từ nút liền kề nút i. Trong bài toán tìm đường đi tối ưu theo quy tắc: một đường là tối ưu nếu và chỉ nếu mọi phân khúc của nó là tối ưu. Vì vậy, nếu tồn tại một đường ngắn nhất từ nút bắt đầu đến nút mục tiêu và tồn tại nút x ở giữa trên đường đó, phân khúc từ nút đầu đến nút x là đường tối ưu từ nút đầu đến nút x. Tương tự, phân khúc từ nút x tới nút mục tiêu cũng đại diện cho đường đi ngắn nhất từ nút x đến nút mục tiêu.

Mô tả giải pháp sử dụng tác tử di động trong cấp phát dữ liệu.

Hệ thống được tổ chức như là đồ thị có hướng với phương pháp sử dụng các tác tử di động thông minh để tìm ra các đường đi ngắn nhất từ một truy vấn được tạo ra ở mỗi nút đến các nút khác liên quan đến truy vấn đó.

Có nhiều giải pháp cho cấp phát mảnh dữ liệu trong các mô hình hệ thống cổ điển. Nhưng trong giải pháp sử dụng tác tử di động nó có thể phản ứng với môi trường và thay đổi nó để thực hiện tốt mục tiêu của mình. Vì vậy chúng tôi kết hợp mô hình đồ thị với bài toán tìm đường đi cho các tác tử cũng như với các loại hình học tập như học không giám sát, học tập trung và hợp tác.

Mục tiêu của hệ thống tác tử là tìm đường đi từ cấu hình ban đầu đến cấu hình mục tiêu, có nghĩa là cần phải phân mảnh dữ liệu và để có được sự cân bằng trong đồ thị, như vậy chúng ta có thể có được thời gian đáp ứng hợp lý từ mỗi nút nơi chúng ta có thể bắt đầu một truy vấn.

Chi phí của mỗi cạnh có một ý nghĩa đó là những đáp ứng tốt nhất của hệ thống trong phần lớn các trường hợp, vì vậy khi thực hiện cấp phát dữ liệu cần phải chú ý, ban đầu chúng ta có thể ước chừng, dự đoán, truy cập từ một site tới dữ liệu của site khác để biết thời gian đáp ứng trên một đơn vị thời gian. Chúng tôi xây dựng đồ thị kết hợp chi phí của thời gian chuyển giao đơn vị dữ liệu. Khi thực hiện cấp phát dữ liệu chúng tôi tăng kích thước vật lý của các mảnh với chi phí của chuyển giao từng đơn vị và từ đó có được ma trận khoảng cách trong biểu đồ. Qua đó có thể áp dụng một số thuật toán ma trận để có một số giá trị nhỏ nhất như các thuật toán Floyd-Hu và Floyd-Warshall-Hu.

Thuật toán Floyd-Hu:

Xét ma trận V= (v_{ij}) các giá trị của các cạnh đồ thị. Định nghĩa ma trận V^k, k =1,..., n+1 và V^1=V, và tính V^{k+1} bởi $v_{ij}^{k+1} = \min\{v_{ij}^k, v_{ik}^k + v_{kj}^k\}$

Đầu vào: Đồ thị cho bởi ma trận V

Đầu ra: Ma trận đường đi ngắn nhất giữa các cặp đỉnh i, j

Giai đoạn 1: Khởi tạo

Giai đoạn 2: Các bước lặp

```
for k=1 to n do
        for i=1 to n do
                for j=1 to n do
```

$v_{ij}^{k+1} = \min\{v_{ij}^{k}, v_{ik}^{k} + v_{kj}^{k}\}$

end for

end for

end for

Độ phức tạp của thuật toán là $O(n^3)$, trong đó n là số đỉnh của đồ thị.

Xây dựng hệ thống với tác tử di động lưu trú trên thực tế có thể cung cấp thông tin chính xác về việc chuyển dữ liệu. Tác tử di động có thể học từ hệ thống và cung cấp một số thống kê về số lần xuất hiện truy vấn trên một site cụ thể, đại diện x tỷ lệ phần trăm của chi phí. Một chi phí khác y tỷ lệ phần trăm là thời gian thực quan trọng cho một truy vấn thực thi ở một site. Chi phí này có thể được xác định bởi tác tử vì tác tử có thể di chuyển qua lại giữa các site và hợp tác với các tác tử khác. Chúng ta có thể thực hiện khởi tạo tham số ban đầu và gán lại giá trị cho các biến. Chúng tôi xây dựng hệ thống tác tử như vậy để nó có thể phản ứng và thay đổi bằng cách gán một số giá trị của chính nó để cải thiện đáp ứng.

Hệ thống có thể thay đổi giá trị của x và y hoặc có thể thêm các biến khác để mô tả chi phí tốt hơn. Chi phí cuối cùng liên quan đến một cạnh được thể hiện bởi 3 yếu tố: truy vấn qo xuất hiện, thời gian thực rti và thời gian đáp ứng rmt lại sau mỗi lần truy vấn. Công thức chi phí cho mỗi cạnh sẽ là:

Cost=qo * x + rti * y + rmt * (100 – x – y)

Chi phí này có thể có tính động tự nhiên do hệ thống mạng có thể chậm bởi đường truyền, giao tiếp dữ liệu khác, những thay đổi của cấu trúc mạng hay thời gian thực bắt đầu khởi tạo của truy vấn ở một nút của hệ thống.

Chúng ta có thể áp dụng nhiều thuật toán tìm kiếm đa tác tử thời gian thực, chia vấn đề chi phí trong nhiều mục tiêu con có thể thay đổi trong suốt thời gian tồn tại của tác tử:

$$Total\ Cost = \sum_{\forall q_i \in Q} QO_i * x + \sum_{\forall S_k \in S} \sum_{\forall F_j \in F} RMT_{jk} * (100 - x - y) + \sum_{\forall q_i \in Q} RTI_{q_i} * y$$

Trong đó: Total Cost là tổng chi phí, Q là tập các truy vấn, S là tập các site và F là tập các mảnh dữ liệu, x và y là tỷ lệ của các yếu tố liên quan đến chi phí như số lần truy vấn của query ở một site và thời gian thực để thực thi truy vấn ở một site.

Ví dụ 8: Từ ví dụ 1 và ví dụ 2, giả sử 4 mảnh ngang của quan hệ EMP được cấp phát trên 4 site lần lượt mảnh EMP1 trên site S_1, EMP2 trên site S_2, EMP3 trên site S_3, EMP4 trên site S_4 và chiến lược phân tán là chia nhỏ dữ liệu, giả sử một yêu cầu "Cho biết thông tin của các nhân viên gồm ENO, ENAME có TITLE là Elec. Eng hoặc Programmer". Câu truy vấn SQL tương ứng là:

```
SELECT  ENO, ENAME
FROM    EMP
WHERE  TITLE= "Elec. Eng" OR TITLE= "Programmer"
```

Giả sử trả lời truy vấn tại S_3, gọi x, y, z là số byte dữ liệu được truyền lần lượt từ site S_1, S_2, S_4 về S_3, theo mô hình chi phí truyền thống thì tổng chi phí thực hiện là: Total_cost=C_{CPU} * #instr + $C_{I/O}$ * #I/OS + C_{MSG} * #msgs + C_{TR} * #bytes, trong đó, hai thành phần chi phí của một lệnh CPU (C_{CPU}) và chi phí của một xuất nhập đĩa ($C_{I/O}$) là các chi phí địa phương, hai thành phần chi phí (C_{MSG}, C_{TR}) là chi phí truyền thông để chuyển số byte dữ liệu từ trạm này đến trạm khác và giả sử hai thành phần chi phí này được biểu thị theo đơn vị thời gian, nên tổng chi phí truyền dữ liệu từ site S_1, S_2, S_4 về S_3 là: Total_cost= $3C_{MSG}$ + C_{TR} * (x + y + z), như vậy chi phí này sẽ lớn khi x, y và z lớn. Vì truyền dữ liệu có thể được thực hiện song song nên thời gian trả lời của truy vấn là: Response_time = max{C_{MSG} + C_{TR} * x, C_{MSG} + C_{TR} * y, C_{MSG} + C_{TR} * z}, thời gian trả lời tối thiểu đạt được nếu tăng mức độ xử lý song song, tuy nhiên không có nghĩa sẽ được tổng chi phí thấp mà có thể ngược lại, tổng chi phí có thể tăng do có nhiều xử lý cục bộ và truyền song song.

Theo giải pháp sử dụng tác tử di động, tác tử được khởi tạo tại site S_3, tác tử di chuyển giữa các site và mang yêu cầu tới site S_1 và S_4 xử lý truy vấn tại 2 site này, sau đó chuyển kết quả về site S_3 lúc này chi phí truyền dữ liệu là: Total_cost= $2C_{MSG}$ + C_{TR} * (x' + z'), trong đó x', z' lần lượt là số byte dữ liệu thỏa điều kiện truy vấn, như vậy chi phí truyền dữ liệu thấp hơn mô hình truyền thống do chỉ truyền dữ liệu là kết quả của truy vấn. Thời gian trả lời của truy vấn nếu thực hiện truyền dữ liệu song song là: Response_time = max{C_{MSG} + C_{TR} * x', C_{MSG} + C_{TR} * z'}, so với mô hình truyền thống thời gian trả lời của truy vấn theo giải pháp tác tử di động cũng thấp hơn.

V. KẾT LUẬN

Sử dụng tác tử di động có thể cung cấp giải pháp tốt trong cấp phát dữ liệu cho hệ thống phân tán. Lợi ích của việc sử dụng tác tử là làm tăng tính tự trị và tăng tính linh hoạt của hệ thống, đồng thời tác tử di động có thể phản ứng lại các thay đổi của hệ thống, thêm vào đó tác tử di động làm giảm đáng kể sự tác động của con người vào quản trị hệ thống. Bài báo đã trình bày giải pháp phân mảnh và cấp phát dữ liệu trong CSDL phân tán sử dụng tác tử di động làm

tăng hiệu suất hệ thống và giảm chi phí truyền dữ liệu so với giải pháp truyền thống. Trong thời gian tới chúng tôi tiến hành thực nghiệm trên các bộ dữ liệu khác nhau nhằm đánh giá giải pháp đã đề xuất, đồng thời tiếp tục nghiên cứu và phát triển áp dụng tác tử di động vào tối ưu hóa truy vấn CSDL phân tán.

TÀI LIỆU THAM KHẢO

[1] Peter Braun, Wilhelm Rossak, Mobile Agents – Basis concepts, Mobility models and the Tracy toolkit, Morgan Kaufmann Publishers, USA, 2005.

[2] M. Tamer Özsu, Patrick Valduriez, Principles of Distributed Database Systems, Springer, 2011.

[3] Shahidul Islam Khan, Dr. A. S. M. Latiful Hoque, A New Technique for Database Fragmentation in Distributed Systems, International Journal of Computer Applications, Vol 5, 2010.

[4] Nicoleta, Magdalena Iacob, Fragmentation and Data Allocation in the Distributed Environments, Annals of the University of Craiova, Mathematics and Computer Science Series, Vol.38, 2011.

[5] Prof. Y M Naik, Shilpa Tarihal, Roopali Swami, shwini Purandare, Kiran Adike, Distributed Information Retrieval Using Mobile Agent, American International Journal of Research in Science, Technology, Engineering & Mathematics, pp. 182-185, USA, 2013.

[6] A. Suganya, R. Kalaiselvi, Efficient Fragmentation and Allocation in Distributed Databases, International Journal of Engineering Research & Technology, Vol.2, 2013.

[7] Vivek N. Waghmare, Snehal D. Patkar, Pranali B. Patil, An Agent Based Mobile Transaction and Disconnection Management System, International Journal of Emerging Technology and Advanced Engineering, Vol. 5, 2015.

[8] Van Nghia Luong, Ha Huy Cuong Nguyen, Van Son Le, An improvement on fragmentation in Distribution Database Design Based on Knowledge-Oriented Clustering Techniques, International Journal of Computer Science and Information Security, Vol. 13, 2015.

FRAGMENTATION AND ALLOCATION OF DISTRIBUTED DATA BY MOBILE AGENT

Tran Dinh Toan, Nguyen Mau Han

ABSTRACT *— In this paper, we introduce solutions fragmented data and data allocation using mobile agent in distributed database systems, the use of mobile agent in order to increase the autonomy, flexibility and responsiveness to change of the system. We focus on the fragmentation and data allocation for the purpose of optimized data allocation, thereby, to reduce the cost of data transmission, improve system performance.*

Keywords *— Distributed database, fragmentation, allocation, mobile agents, optimization.*