

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

BÙI THỊ LỰA

TỐI ƯU HOÁ TRUY VẤN
CƠ SỞ DỮ LIỆU SONG SONG

Chuyên ngành: KHOA HỌC MÁY TÍNH
Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

ĐÀ NẴNG – Năm 2010

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: PGS.TSKH. TRẦN QUỐC CHIẾN

Phản biện 1: TS. NGUYỄN MẬU HÂN

Phản biện 2: TS. NGUYỄN TRẦN QUANG VINH

Luận văn được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp
thạc sĩ Kỹ thuật họp tại Đại học Đà Nẵng vào ngày 14 tháng 10 năm
2010.

Có thể tìm hiểu luận văn tại:

- Trung tâm Thông tin-Học liệu, Đại học Đà Nẵng
- Trung tâm học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. Lý do chọn đề tài

Đề khai thác các khả năng của các máy CSDL song song nhằm đạt được hiệu quả tốt nhất có thể. Cùng với việc xử lý các truy vấn, tối ưu hoá truy vấn trên môi trường đa xử lý là một vấn đề quan trọng dẫn tới sự thành công trong kỹ thuật CSDL song song. Nó quyết định tốc độ hồi đáp nhanh nhất có thể cho các truy vấn, đó là lý do chính yếu để sử dụng các hệ thống song song.

2. Mục đích nghiên cứu

- Nghiên cứu một số kiến trúc CSDL song song, các phương pháp song song hoá dữ liệu nhằm giải quyết vấn đề bế tắc vào ra thường gặp trong các hệ CSDL song song.

- Nghiên cứu phương pháp tối ưu hoá hai pha và các thuật toán trong giai đoạn đầu của mô hình tối ưu hoá hai pha nhằm biểu diễn lại câu truy vấn thành cây truy vấn có chú giải.

3. Đối tượng và phạm vi nghiên cứu

- Nghiên cứu một số kiến trúc CSDL song song và các chiến lược song song hoá dữ liệu.

- Nghiên cứu quá trình tối ưu hoá truy vấn song song: Nghiên cứu mô hình tối ưu hoá truy vấn cho CSDL song song, và các thuật toán liên quan đến bài toán tối ưu hoá truy vấn trên môi trường xử lý song song.

4. Phương pháp nghiên cứu

- Thu thập và phân tích các tài liệu và thông tin liên quan đến đề tài.

- Lựa chọn, đề xuất phương hướng giải quyết vấn đề.

- Kiểm tra, thử nghiệm và đánh giá kết quả

5. Ý nghĩa khoa học và thực tiễn của đề tài

- Tối ưu hoá truy vấn song song quyết định tốc độ hồi đáp nhanh nhất có thể có cho các truy vấn, qua đó giúp việc tổ chức và khai thác các cơ sở dữ liệu trên môi trường đa xử lý có hiệu quả tốt hơn.

- Các thuật toán được nghiên cứu dựa trên kỹ thuật quy hoạch động, có tính đến các chi phí phân bố lại, là một đóng góp cho giai đoạn tối ưu hoá truy vấn song song.

6. Cấu trúc của luận văn

Luận văn gồm 4 chương:

Chương 1: Chương này sẽ giới thiệu qua một số hoạt động của bài toán tối ưu hoá truy vấn trong các môi trường: Tập trung, phân tán và song song.

Chương 2: Trình bày những vấn đề về các kiến trúc CSDL song song và các chiến lược song song hoá dữ liệu.

Chương 3: Trình bày một mô hình tối ưu hoá truy vấn cho CSDL song song để thấy sự khác nhau giữa tối ưu hoá truy vấn cho CSDL song song và CSDL tuần tự cổ điển.

Chương 4: Trình bày một số minh họa cho các thuật toán đã được trình bày trong chương 3.

Kết thúc luận văn này là phần kết luận, tóm lược lại những vấn đề đã trình bày và một số hướng phát triển trong tương lai.

CHƯƠNG 1: TỐI ƯU HOÁ TRUY VẤN TRONG CÁC MÔI TRƯỜNG: TẬP TRUNG, PHÂN TÁN VÀ SONG SONG

1.1. TỐI ƯU HOÁ TRUY VẤN TRONG CƠ SỞ DỮ LIỆU TẬP TRUNG

1.1.1. Bài toán xử lý vấn tin tập trung

1.1.2. Ngôn ngữ

1.1.3. Các kiểu tối ưu hoá

1.1.4. Thời điểm tối ưu hoá

1.1.5. Số liệu thống kê

1.2. TỐI ƯU HOÁ TRUY VẤN TRONG CƠ SỞ DỮ LIỆU PHÂN TÁN

1.2.1. Bài toán xử lý vấn tin phân tán

1.2.2. Mô hình chi phí phân tán

1.2.3. Mô tả đặc trưng của thể xử lý vấn tin phân tán

Thể xử lý vấn tin phân tán cũng có các đặc trưng như đối với xử lý vấn tin tập trung như *ngôn ngữ, các kiểu tối ưu hoá, thời điểm tối ưu hoá, số liệu thống kê*.

Ngoài ra, thể xử lý vấn tin phân tán còn có các đặc trưng sau:

1.2.3.1. Vị trí quyết định

1.2.3.2. Tận dụng cấu hình mạng

1.2.3.3. Tận dụng các mảnh nhân bản

1.2.3.4. Sử dụng các nối nửa

1.2.4. Các tầng xử lý vấn tin

1.2.4.1. Phân rẽ vấn tin

1.2.4.2. Cục bộ hoá dữ liệu

1.2.4.3. Tối ưu hoá vấn tin toàn cục

1.2.4.4. Tối ưu hoá vấn tin cục bộ

1.3. TỐI ƯU HOÁ TRUY VẤN SONG SONG

Một yếu tố dẫn đến sự thành công của các hệ cơ sở dữ liệu đa xử lý này là tính hiệu quả của bộ tối ưu hoá. Chức năng chính của bộ tối ưu là tìm một chiến lược thực thi tốt nhất cho câu truy vấn SQL đầu vào. Đầu ra của bộ tối ưu là một lịch trình bao gồm các truy vấn đại số và thứ tự thực hiện của chúng được xác định trên các mảnh dữ liệu ở các nút cùng với các phép toán truyền thông có sẵn. Các phương pháp tối ưu truy vấn trong môi trường đa xử lý dưới dạng những thuật giải Heuristic đã được áp dụng trong các hệ CSDL song song.

- Phương pháp cổ điển

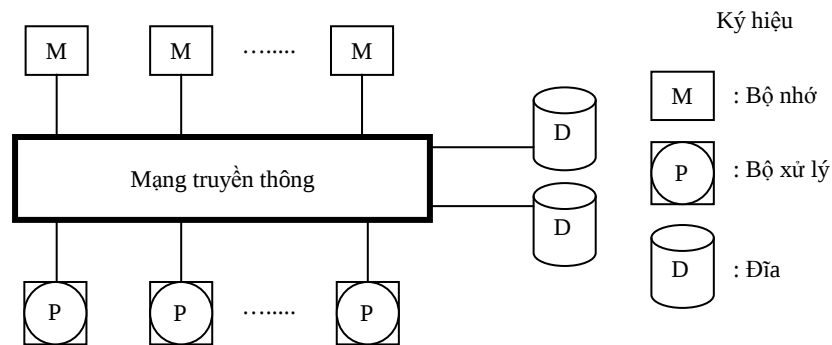
- Phương pháp quy hoạch động

- Phương pháp tối ưu hoá hai pha: Kế thừa những ưu điểm của chiến lược tối ưu trong xử lý tuần tự, phương pháp này được chia thành hai nhiệm vụ con: Đầu tiên, đưa ra một phương án thực hiện mà chưa phải chú ý đến cách phân phối công việc cho các bộ xử lý. Sau đó, đưa ra một lịch biểu và phương án tối ưu thực hiện song song các phép toán có được từ nhiệm vụ thứ nhất. Điều thuận tiện của chiến lược tối ưu hoá hai pha là mềm dẻo, đơn giản và tận dụng được những kết quả về tối ưu hoá trong môi trường xử lý tuần tự.

CHƯƠNG 2: KIẾN TRÚC CÁC HỆ CƠ SỞ DỮ LIỆU SONG SONG

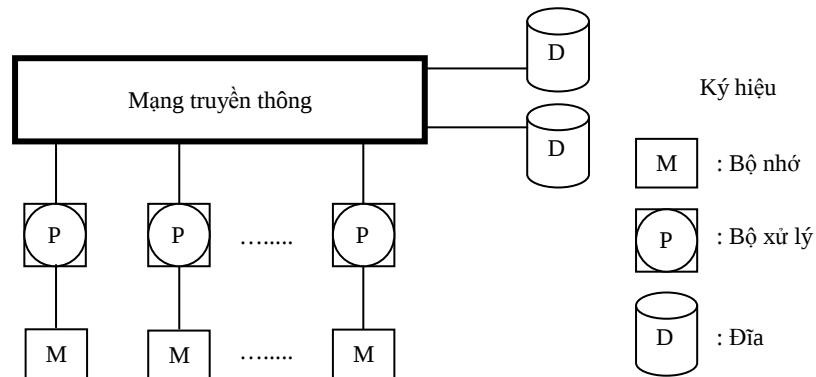
2.1. CÁC KIẾN TRÚC CỦA HỆ THỐNG MÁY TÍNH SONG SONG

2.1.1. Kiến trúc mọi thứ dùng chung (shared everything)



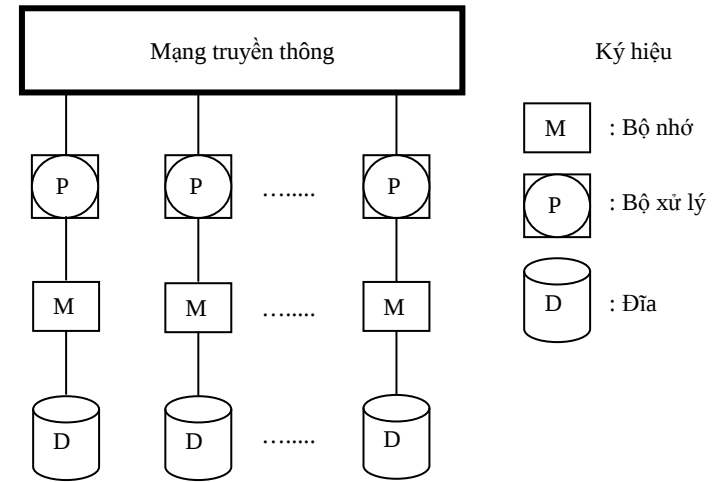
Hình 2.1: Kiến trúc mọi thứ dùng chung

2.1.2. Kiến trúc dùng chung đĩa (shared disk)



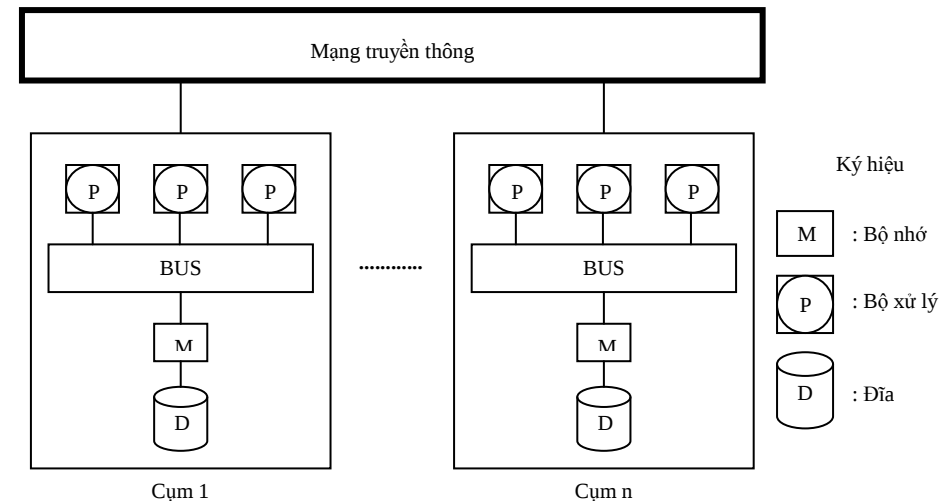
Hình 2.2: Kiến trúc dùng chung đĩa

2.1.3. Kiến trúc không chia sẻ (shared nothing)



Hình 2.3: Kiến trúc không chia sẻ

2.1.4. Kiến trúc phân cấp



Hình 2.4: Kiến trúc phân cấp

2.2. CÁC KỸ THUẬT PHÂN HOẠCH DỮ LIỆU TRONG CSDL SONG SONG

Nhằm giải quyết vấn đề bế tắc vào ra thường gặp trong các hệ CSDL song song, ngoài việc áp dụng một kiến trúc phần cứng thích hợp, người ta còn phải tiến hành phân hoạch dữ liệu một cách hợp lý cho các bộ xử lý.

2.2.1. Phân hoạch theo vòng tròn Robin

2.2.2. Phân hoạch theo hàm băm

2.2.3. Phân hoạch theo khoảng

2.2.4. Phân hoạch theo mảng nhiều chiều

2.3. CÁC CƠ CHẾ XỬ LÝ SONG SONG

Các phép truy vấn trong mô hình quan hệ thực sự phù hợp với việc thực hiện song song. Truy vấn quan hệ thực chất là các phép toán quan hệ thực hiện trên các dòng dữ liệu có cùng cấu trúc. Hơn nữa, kết quả của mỗi phép toán là một quan hệ nên ta có thể kết hợp các phép toán thành các dòng dữ liệu song song. Có hai loại dòng dữ liệu song song: song song đường ống và song song phân hoạch. Phương pháp tiếp cận dòng dữ liệu song song cho phép sử dụng các thủ tục tuần tự sẵn có để thực hiện các phép toán đã có một cách song song mà không phải xây dựng các phép toán song song mới.

2.3.1. Song song liên truy vấn

Song song liên truy vấn là thực hiện nhiều truy vấn cùng một lúc bằng cách lập lịch thực hiện cho các toán tử của các truy vấn đó.

2.3.1.1. Lập lịch trên cơ sở cạnh tranh

2.3.1.2. Lập lịch theo phương án

2.3.2. Song song nội truy vấn

Song song nội truy vấn là dạng song song hoá thi hành song song một truy vấn đơn trên nhiều bộ xử lý và đĩa.

2.3.2.1. Song song độc lập

2.3.2.2. Song song đường ống

2.3.3. Song song nội toán tử

Song song nội toán tử là thực hiện một phép toán quan hệ bằng cách dùng nhiều bộ xử lý.

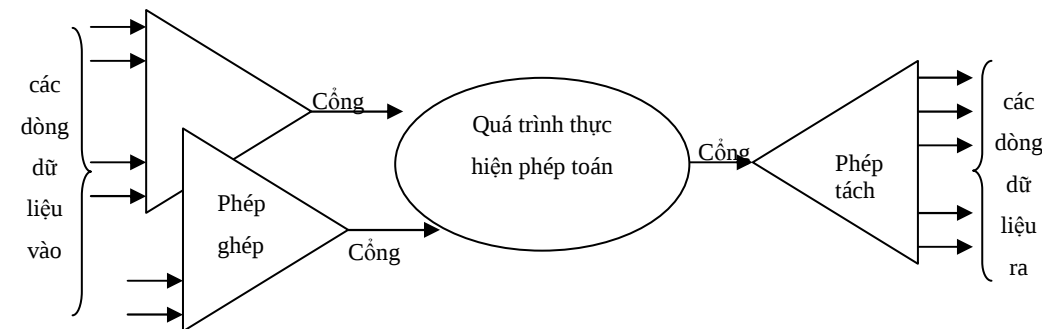
2.4. CÁC PHÉP TOÁN SONG SONG

2.4.1. Phép ghép

Phép ghép dùng để kết hợp nhiều dòng dữ liệu song song thành một dòng dữ liệu đơn.

2.4.2. Phép tách

Khi thực hiện một quy trình song song nhiều giai đoạn, một dòng dữ liệu đơn phải được tách thành nhiều dòng dữ liệu độc lập.



Hình 2.7: Ghép các dòng dữ liệu vào và tách các dòng dữ liệu ra của một phép toán

2.4.3. Các thuật toán xử lý các phép gộp nhóm

Một hàm gộp nhóm SQL là một hàm thao tác trên các nhóm mẫu tin có cùng một tính chất nào đó.

2.4.3.1. Thuật toán trộn tập trung CM (Centralized Merging)

2.4.3.2. Thuật toán trộn phân tán DM (Distributed Merging)

2.4.3.3. Thuật toán phân hoạch lại Rep (Repartitioning)

2.4.3.4. Các thuật toán thực hiện phép kết nối tự nhiên

CHƯƠNG 3: TỐI ƯU HOÁ TRUY VẤN SONG SONG

3.1. MÔ HÌNH CHI PHÍ CỦA BỘ TỐI ƯU HOÁ TRUY VẤN

Chi phí thực hiện một phương án tối ưu của một câu truy vấn song song được xác định bởi ba thành phần: tổng công việc TW (Total Work), thời gian trả lời RT (Response Time) và chi phí không gian nhớ MC (Memory Consumption). Hàm chi phí là tổ hợp của hai thành phần đầu, thành phần thứ ba cho biết kích thước bộ nhớ cần cho việc thực thi phương án.

3.1.1. Các thông kê CSDL

3.1.2. Lực lượng của các kết quả trung gian

3.1.2.1. Phép chọn

3.1.2.2. Phép chiếu

3.1.2.3. Phép kết nối

3.1.2.4. Phép nửa kết nối

3.1.2.5. Phép hợp

3.1.2.6. Phép trừ

3.1.3. Chi phí song song

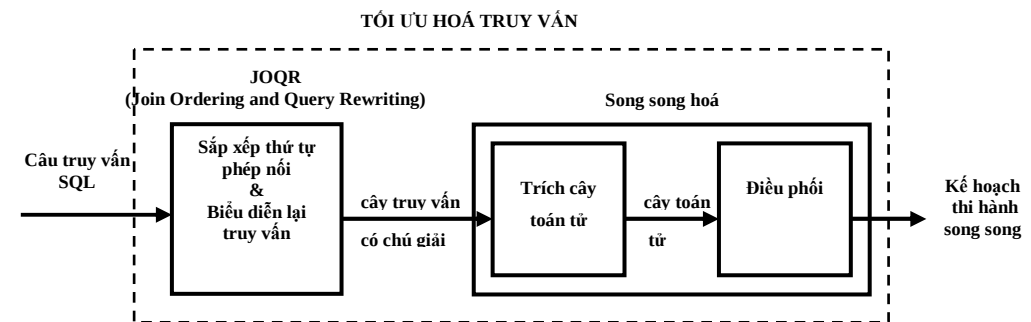
3.1.4. Chi phí khởi động

3.1.5. Chi phí truyền thông

3.1.6. Ước lượng chi phí song song

3.2. MÔ HÌNH TỐI ƯU HOÁ TRUY VẤN CHO CSDL SONG SONG

Trong phần trình bày này, chúng ta sẽ mô tả chi tiết quá trình tối ưu hoá truy vấn song song bằng mô hình tối ưu hoá truy vấn hai pha để cực tiểu hoá thời gian hồi đáp. Pha đầu tiên áp dụng thủ thuật cực tiểu hoá tổng khối lượng công việc, trong pha sau sử dụng hai thủ thuật phân chia công việc lên nhiều bộ xử lý. Việc chia bài toán thành hai pha cho phép giảm độ phức tạp khi tối ưu hoá câu truy vấn song song.



Hình 3.2: Các giai đoạn của quá trình tối ưu hoá truy vấn hai pha

3.2.1. Cây truy vấn có chú giải

Cây truy vấn có chú giải là dạng trình bày truyền thống của phương án thi hành một câu truy vấn SQL. Nó mã hoá những lựa chọn mang tính thủ tục như thứ tự thực hiện mỗi phép toán, phương pháp tính toán mỗi toán tử. Mỗi nút của cây đại diện cho một (hay nhiều) phép toán quan hệ, mỗi nút lá đại diện cho một quan hệ toán hạng. Những ghi chú trên mỗi nút mô tả cách thức nó được thực hiện chi tiết như thế nào.

3.2.2. Cây toán tử

Cây toán tử dùng để mô tả các phép toán song song thực hiện cây truy vấn cũng như các ràng buộc về thời gian giữa chúng. Các nút của một cây toán tử biểu diễn các toán tử và các đoạn mã lệnh đơn. Các cạnh tượng trưng cho các dòng dữ liệu, hướng chỉ của mỗi cạnh thể hiện ràng buộc thời gian giữa các toán tử.

3.3. CỰC TIỂU HOÁ KHỐI LƯỢNG TRUY VẤN SONG SONG

3.3.1. Mô hình cực tiểu phí tổn truyền thông

Các phương pháp song song hoá phân hoạch vốn khai thác sự phân hoạch các quan hệ theo chiều ngang, các phương pháp này thường đòi hỏi dữ liệu phải được phân hoạch lại, do đó sẽ làm phát sinh thêm chi phí truyền thông.

3.3.1.1. Phân hoạch

Định nghĩa 3.1: Một phân hoạch là một cặp (a, h) , trong đó a là một thuộc tính và h là một hàm, hàm này ánh xạ một giá trị của a thành một giá trị không âm.

Định nghĩa 3.2: Toán tử một ngôi f là khả năng phân hoạch ứng với phân hoạch α nếu $f(S) = f(S_0) \cup f(S_1) \cup \dots \cup f(S_k)$. Toán tử hai ngôi f là khả năng phân hoạch ứng với phân hoạch α nếu $f(S, T) = f(S_0, T_0) \cup f(S_1, T_1) \cup \dots \cup f(S_k, T_k)$.

Định nghĩa 3.3: Một phép toán được gọi là cảm thuộc tính nếu nó chỉ khả phân hoạch trên các phân hoạch sử dụng một thuộc tính nhất định. Ngược lại, nếu phép toán khả phân hoạch trên mọi phân hoạch thì gọi là bất cảm thuộc tính.

3.3.1.2. Chi phí phân hoạch lại

Trường hợp các toán tử sử dụng các phân hoạch khác nhau, các bộ đầu ra của toán tử này là đầu vào của toán tử kia thì dữ liệu cần được phân hoạch lại theo đúng yêu cầu của toán tử sau. Chúng ta nhận thấy rằng, một yếu tố quan trọng ảnh hưởng đến chi phí thực hiện truy vấn là thuộc tính được sử dụng để phân hoạch.

3.3.1.3. Bài toán tối ưu hoá

Định nghĩa 3.4: Màu của một nút trên cây truy vấn là một thuộc tính được sử dụng cho việc phân hoạch dữ liệu tại nút đó. Một cạnh giữa nút i và j được gọi là cạnh đa màu nếu i và j được gán hai màu khác nhau.

Trong một cây truy vấn, các nút là toán tử cảm thuộc tính hoặc bất cảm thuộc tính thì được tô màu trước, chúng ta tự do ấn định màu cho các nút không màu còn lại.

Mỗi cạnh e của cây truy vấn chúng ta gán một trọng số C_e để tượng trưng cho chi phí phân hoạch lại. Chi phí này chỉ được tính khi cạnh này là đa màu do đó tổng chi phí phân hoạch lại chính là tổng

trọng số của các cạnh đa màu. Bài toán tối ưu được phát biểu dưới dạng bài toán tô màu như sau:

“Cho trước một cây truy vấn $T=(V, E)$, gọi C_e là trọng số của cạnh $e \in E$, giả sử một số nút trong của V được tô màu trước. Hãy tô màu cho các nút còn lại để cực tiểu hoá tổng trọng số của các cạnh đa màu”.

3.3.2. Các thuật toán tô màu tối ưu cho một cây truy vấn

3.3.2.1. Bài toán tô màu đơn giản

Bài toán tô màu cho một cây nói chung có thể thu lại thành bài toán tô màu một tập các cây đơn giản, trong đó tất cả các nút trong là chưa tô màu và tất cả các nút lá là đã được tô màu trước. Thủ tục Đơn Giản Hoá sau đây thực hiện đơn giản hoá cây truy vấn.

Thuật toán 3.1. Thủ tục ĐơnGiảnHoá (Simplify)

Input Cây truy vấn

Output Cây truy vấn thu gọn các lá không màu

1. **While** tồn tại nút lá không màu l với nút cha m **do**

2. Thu gọn l về m ;

3. **While** tồn tại nút trong có màu m bậc d **do**

4. Tách m thành d nút bản sao, mỗi nút bản sao được liên thông với phần còn lại bằng một cạnh;

3.3.2.2. Thuật toán tham lam trong trường hợp các màu tô trước khác nhau

Định nghĩa 3.5: Một nút là *nút mẹ* nếu và chỉ nếu tất cả các nút kề là nút lá với tối đa một ngoại lệ. Trong trường hợp này, nút lá được gọi là nút con của nút mẹ.

Bổ đề 3.1: Tồn tại một phép tô màu tối ưu cắt các cạnh $e_1, e_2, \dots, e_{d-1}$.

Bổ đề 3.2: Tồn tại phép tô màu tối ưu trong đó có thu gọn e_2 .

Lưu ý: Chỉ có các nút lá được tô màu luôn được giữ vững sau mỗi khi áp dụng các bổ đề cắt/thu gọn cạnh. Do đó, các bổ đề này có thể áp dụng cho bất kỳ một cây không tầm thường nào. Do việc áp dụng bổ đề làm giảm số cạnh, việc áp dụng lặp lại nhiều lần sẽ dẫn đến tập những cây tầm thường. Nhận xét này dẫn đến thuật toán Tô Màu dưới đây trong việc tìm phép tô màu tối ưu.

Thuật toán 3.2. Thuật toán ToMau

Input Cây truy vấn đơn giản, các nút lá tô màu trước khác nhau.

Output Cây được tô màu.

1. **While** tồn tại nút mẹ m có bậc tối thiểu là 2 **do**

2. Cho các cạnh từ nút mẹ m đến d nút con là $e_1, \dots, e_d$ sao cho $C_{e_1} \leq \dots \leq C_{e_d}$;

3. **If** $d > 1$ **Then** cắt $e_1, \dots, e_{d-1}$

4. **Else** Gọi e_p là cạnh từ m đến nút cha của nó;

5. **If** $C_{e_p} < C_{e_1}$ **Then** thu gọn e_1 **Else** thu gọn e_p ;

6. **End While**;

7. Tô màu các cây tầm thường;

Do mỗi vòng lặp làm giảm số cạnh, thời gian thực hiện thuật toán là tuyến tính theo số cạnh.

3.3.2.3. Thuật toán tách màu

Các ký hiệu:

Gọi $Optc(i, A)$ là chi phí cực tiểu của phép tô màu cây con gốc i sao cho i được tô màu A .

Gọi $Opt(i)$ là chi phí cực tiểu để tô màu cây con gốc i , bắt chấp màu của i trước đó. Nghĩa là, $Opt(i) = \min_a Optc(i, a)$, a là một màu bất kỳ.

Bổ đề 3.3: Chi phí cực tiểu của phép tô màu cây con gốc i sao cho i có màu A được xác định bởi:

$$Optc(i, A) = \begin{cases} \infty & \text{nếu } i \text{ được tô màu trước khác màu } A \\ 0 & \text{nếu } i \text{ là nút lá, không màu hoặc màu } A \\ \sum_{1 \leq j \leq k} \min [Optc(\alpha_j, A), c_j + Opt(\alpha_j)] & \text{các trường hợp khác} \end{cases}$$

Bổ đề 3.4: Nếu i nhận màu A trong một phép tô màu tối ưu nào đó, thì cũng tồn tại một phép tô màu tối ưu khác sao cho nút con α_j của i có màu A nếu $Optc(\alpha_j, A) \leq c_j + Opt(\alpha_j)$, trong trường hợp ngược lại thì sẽ có màu a bất kỳ nếu $Optc(\alpha_j, a) = Opt(\alpha_j)$.

Các bổ đề 3.3 và 3.4 dẫn đến thuật toán Tách Màu dưới đây. Gọi C là tập màu đã được dùng cho các nút được tô màu trước.

Thuật toán 3.3. Thuật toán TachMau (Color Split)

Input Cây truy vấn T , tập màu C .

Output Phép tô màu tối ưu.

1. **For** mỗi nút i theo thứ tự hậu tố **do** bước 2
2. **For** mỗi màu $a \in C$ **do** bước 3 và bước 4
3. Tính $Optc(i, a)$ bằng cách sử dụng bổ đề 3.3;
4. $Opt(i) = \min_a Optc(i, a)$;
5. Tìm $a \in C$ sao cho $Optc(r, a) = Opt(r)$, trong đó r là gốc.
6. Màu(r)= a ;
7. **For** mỗi nút α_j không phải là gốc theo thứ tự tiền tố **do** bước 8 đến bước 10.
8. Gọi i là nút cha của α_j ; c_j là trọng số của cạnh nối i và α_j ;

9. **If** $Optc(\alpha_j, \text{Màu}(i)) \leq c_j + Opt(\alpha_j)$ **then** Màu(α_j) = Màu(i)

10. **Else** Màu(α_j) = $a \in C$ sao cho $Optc(\alpha_j, a) = Opt(\alpha_j)$;

End.

Thuật toán Tách Màu không yêu cầu tất cả các nút lá của cây đầu vào phải tô màu trước mà đi tìm phép tô màu tối ưu cho mọi cây. Thuật toán này có thời gian thực hiện $O(n|C|)$.

3.3.2.4. Các mở rộng: sử dụng tập màu

Bổ đề 3.5: Chi phí cực tiểu $Optc(i, A)$ của phép tô màu cây con gốc i sao cho i nhận màu A được xác định bởi công thức.

$$Optc(i, A) = \begin{cases} \infty & \text{nếu } A \text{ không ở trong tập màu tô trước của } i \\ 0 & \text{nếu } i \text{ là nút lá, không màu hoặc màu } A \\ \sum_{1 \leq j \leq k} \min [Optc(\alpha_j, A), c_j + Opt(\alpha_j)] & \text{các trường hợp khác} \end{cases}$$

3.3.3. Mô hình cho các phương pháp và đặc tính vật lý

3.3.3.1. Chi phí của cây truy vấn có chú giải

Các ký hiệu:

R^s là bảng R với tính chất thống kê s .

$recolor(R^s, c_{old}, c_{new})$ là chi phí tô màu lại bảng R từ màu c_{old} thành màu c_{new} .

$inpCol(s, A, j)$ là mẫu màu cần cho chiến lược s với đầu vào thứ j để được kết xuất mẫu màu A .

$StrategyCost(s, R_1^s, \dots, R_k^s)$ là chi phí thực hiện chiến lược tính toán s trên các bảng R_i .

Giả sử gốc của cây truy vấn T sử dụng chiến lược s có màu đầu ra là a . Gọi $C_j = inpCol(s, a, j)$ là màu của đầu vào thứ j mà

chiến lược s đòi hỏi. Giả sử T có k cây con $T_1, T_2, \dots, T_k$ sao cho T_j tạo ra bảng R_j với màu c_j . Khi đó chi phí của cây truy vấn chú giả T được xác định như sau:

Công thức (3.8):

$$\text{cost}(T) = \text{StrategyCost}(s, R_1^s, \dots, R_k^s) + \sum_{j=1}^k \text{recolor}(R_j^s, c_j, c'_j) + \sum_{j=1}^k \text{cost}(T_j)$$

3.3.3.2. Thuật toán tách màu mở rộng

Định nghĩa 3.6: $\text{OptcStrategy}(i, A)$ là chiến lược thực hiện chi phí cực tiểu $\text{Optc}(i, A)$ (nếu có nhiều chiến lược thì chọn lấy một chiến lược). $\text{OptcStrategy}(i, A)$ không định nghĩa cho nút lá.

Định nghĩa 3.7: $\text{Strategies}(i, A)$ là tập hợp những chiến lược có thể áp dụng cho phép toán được biểu diễn bởi nút i và ràng buộc về đầu vào-đầu ra của nó cho phép A như một màu đầu ra.

Bổ đề dưới đây là trường hợp tổng quát của bổ đề 3.5.

Bổ đề 3.6: Cho cây truy vấn T , i là một nút trên T . Chi phí tô màu tối thiểu của cây con gốc i sao cho i có màu đầu ra A , ký hiệu $\text{Optc}(i, A)$, được xác định bởi các công thức dưới đây.

Trường hợp i là nút lá,

$$\text{Optc}(i, A) = \begin{cases} 0 & \text{nếu } i \text{ có màu tương thích với } A \\ \infty & \text{trường hợp khác} \end{cases}$$

Trường hợp i không phải là nút lá, $\text{Optc}(i, A)$ được tính theo công thức truy hồi:

$$\text{Optc}(i, A) = \min_{s \in S} [\text{StrategyCost}(s, R_1^s, \dots, R_k^s) + \sum_{j=1}^k \min_{c \in C} [\text{Optc}(\alpha_j, c) + \text{recolor}(R_j^s, c, \text{inpCol}(s, A, j))]]$$

Trong đó, $S = \text{Strategies}(i, A)$ và $\text{OptcStrategy}(i, A)$ là chiến lược nào đó mà thực hiện chi phí cực tiểu $\text{Optc}(i, A)$. C là một tập

màu mà các cây con gốc tại α_j có thể nhận. $\text{StrategyCost}(s, R_1^s, \dots, R_k^s)$ là chi phí thực hiện chiến lược tính toán s trên các bảng R_i . Thuật toán Tách Màu Mở Rộng dưới đây để tính Optc và OptcStrategy .

Thuật toán 3.4. Thuật toán TachMauMoRong (*Extended Color Split*)

Input Cây truy vấn T với màu của các nút lá, tập màu C .

Output Cây với chiến lược tô màu có chi phí tối thiểu.

1. **For** mỗi nút i theo thứ tự hậu tố **do** bước 2
2. **For** mỗi màu $a \in C$ **do** dùng bổ đề 3.6 để tính $\text{Optc}(i, a)$ và $\text{OptcStrategy}(i, a)$;
3. Xem r là gốc và a là màu sao cho $\text{Optc}(r, a) \leq \text{Optc}(r, c)$ với mọi màu $c \in C$;
4. Màu tối ưu cho r là a và chiến lược tối ưu là $\text{OptcStrategy}(r, a)$;
5. **For** mỗi nút không là nút gốc theo thứ tự tiền tố **do** bước 6
6. Tính màu tối ưu và các chiến lược bằng cách áp dụng bổ đề 3.6 “ngược”;

End.

Thuật toán này có thời gian thực hiện trong trường hợp xấu nhất là $nS|C|^2$ trong đó S là số chiến lược, $|C|$ là số màu được phép, còn n là số nút của cây.

3.3.4. Mô hình cho thứ tự phép kết nối

3.3.4.1. Thứ tự thực hiện kết nối bỏ qua các tính chất vật lý

Định nghĩa 3.8: Cây kết nối (*join tree*) là một cây truy vấn có chú giải, trong đó, tất cả các nút trong biểu diễn các phép kết nối và các nút lá biểu diễn các bảng.

Bổ đề 3.7: Nếu $OptPlan(Q)=[s, T_l, T_r]$ và $Q = Q_l \cup Q_r$ thì $OptPlan(Q_l)=T_l$ và $OptPlan(Q_r)=T_r$, trong đó T_l và T_r lần lượt là kết quả thu được từ nhánh truy vấn Q_l và Q_r .

Thuật toán 3.5. Thuật toán ĐỉnhThuTuPhepKet (Join Ordering)

Input Câu truy vấn SPJ (chọn-chiều-kết nối) trên các bảng $T=\{T_1, T_2, \dots, T_n\}$.

Output Cây kết nối tối ưu.

1. **For** $i:=1$ **to** n **do** $OptPlan(\{T_i\})=T_i$;
 2. **For** $i:=2$ **to** n **do** bước 3
 3. **For** mỗi $Q \subseteq T$ sao cho $|Q|=i$ **do** bước 4 và bước 5
 4. $BestCost = \infty$;
 5. **For** mỗi $Q_l \neq \emptyset$ và $Q_r \neq \emptyset$ sao cho $Q = Q_r \cup Q_l$ **do** bước 6 và bước 7
 6. Gọi R_l^s và R_r^s là các thống kê của các bảng được tính từ các truy vấn Q_l và Q_r ;
 7. **For** mỗi chiến lược kết nối s **do** bước 8 đến 11
 8. **If** $StrategyCost(s, R_l^s, R_r^s) < BestCost$ **then**
 9. $BestCost = StrategyCost(s, R_l^s, R_r^s)$;
 10. $OptPlan(Q)=[s, OptPlan(Q_l), OptPlan(Q_r)]$;
 11. **End if**
- End.**

Thuật toán này có độ phức tạp $O(3^n)$.

3.3.4.2. Thứ tự thực hiện phép kết nối với các tính chất vật lý

Bổ đề 3.8: $Optc(Q, a)$ thoả mãn hệ thức truy hồi sau:

$$Optc(Q, a) = \min_{Q_l, Q_r} [\min_{s \in S} [StrategyCost(s, Q_l, Q_r) + \min_{a \in C} [Optc(Q_l, a) + recolor(Q_l^s, a, inpCol(s, a, 1))] + \min_{a \in C} [Optc(Q_r, a) + recolor(Q_r^s, a, inpCol(s, a, 2))]]]$$

Trong đó, Q_l và Q_r là tất cả các tập sao cho $Q=Q_l \cup Q_r$, $Q_l \neq \emptyset$, $Q_r \neq \emptyset$, S là tập tất cả các chiến lược tạo nên tính chất vật lý a .

Thuật toán 3.6. Thuật toán ThuTuPhepKetVatLy (Join Ordering With Physical Properties)

Input Câu truy vấn SPJ (chọn-chiều-kết nối) trên các bảng $T=\{T_1, T_2, \dots, T_n\}$.

Output Cây kết nối tối ưu.

1. **For** $i:=1$ **to** n **do** bước 2
2. Tính $Optc(T_i, a)$ theo công thức:

$$Optc(T_i, a) = \begin{cases} 0 & \text{nếu } T_i \text{ có phương pháp truy nhập với tính chất vật lý } a; \\ \infty & \text{trong trường hợp khác;} \end{cases}$$

3. **For** $i:=2$ **to** n **do** bước 4
4. **For** mỗi $Q \subseteq T$ sao cho $|Q|=i$ **do** bước 5 và 6
5. Đặt $Optc(Q, a) = \infty$ cho mỗi tính chất vật lý $a \in C$;
6. **For** mỗi $Q_l \neq \emptyset$ và $Q_r \neq \emptyset$ sao cho $Q = Q_r \cup Q_l$ **do** bước 7 và 8
7. Gọi R_l^s, R_r^s là các tính chất thống kê của các bảng được tính từ các truy vấn Q_l, Q_r ;
8. **For** mỗi tính chất vật lý $a \in C$ **do** bước 9
9. **For** mỗi chiến lược s có thể tạo ra tính chất a **do** bước 10 và 11
10. Đặt $s \text{ cost} = StrategyCost(s, R_l^s, R_r^s)$,
 $a_l' = inpCol(s, a, 1)$ và
 $a_r' = inpCol(s, a, 2)$;

11. **For** mỗi tính chất vật lý $a_l \in C, a_r \in C$
do bước 12 đến bước 16
12. Đặt
 $NewCost = scost + Optc(Q_l, a_l) +$
 $recolor(R_l^s, a_l, a_l') + Optc(Q_r, a_r) + recolor(R_r^s, a_r, a_r')$;
13. **If** $NewCost < Optc(Q, a)$ **then**
14. $Optc(Q, a) = NewCost$;
15. $OptPlan(Q, a) = [s, OptPlan(Q_l, a_l), OptPlan(Q_r, a_r)]$;
16. **End if**
17. **Return** $\min_{a \in C} OptPlan(T, a)$;

3.3.5. Tóm Lại

CHƯƠNG 4: CÀI ĐẶT MINH HOẠ CÁC THUẬT TOÁN

4.1. MINH HOẠ THUẬT TOÁN TÁCH MÀU

4.1.1. Bài toán tô màu cây truy vấn

Đầu vào: Cho trước một cây truy vấn $T=(V, E)$, trong đó V là tập đỉnh và E là tập cạnh. Mỗi cạnh $e \in E$ có một trọng số C_e , một số nút trong V được tô màu trước.

Đầu ra: Cây truy vấn được tô màu sao cho tổng trọng số các cạnh nhiều màu là nhỏ nhất.

4.1.2. Cơ sở tính toán

4.1.3. Kết quả cài đặt

4.2. MINH HOẠ THUẬT TOÁN TÁCH MÀU MỞ RỘNG

4.2.1. Bài toán

Đầu vào: Cho cây truy vấn $T=(V, E)$, trong đó V là tập đỉnh và E là tập cạnh. Các nút lá trong V đã được tô màu trước. Cho O là tập các phép toán tại các nút trong. Mỗi phép toán $f \in O$ có một tập các chiến lược thực hiện S_f . D là tình trạng vật lý của CSDL.

Đầu ra: Chiến lược thực hiện có phí tổn ít nhất, màu đầu ra và các đầu vào cho mỗi nút trong.

4.2.2. Các giải thích

4.2.3. Cơ sở tính toán

4.2.4. Kết quả cài đặt

4.3. MINH HOẠ THUẬT TOÁN THỨ TỰ PHÉP KẾT VẬT LÝ

4.3.1. Bài toán

Đầu vào: Cho cây truy vấn *kết nối-chọn-chiều* lên trên tập các bảng $T = \{T_1, T_2, \dots, T_n\}$. Cho O là tập các phép toán tại các nút trong. Mỗi phép toán $f \in O$ có một tập các chiến lược thực hiện S_f . D là tình trạng vật lý của CSDL T .

Đầu ra: Cây kết nối tối ưu

4.3.2. Cơ sở tính toán

4.3.2. Kết quả cài đặt

4.4. CÔNG THỨC TÍNH CHI PHÍ

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

CSDL trên các máy tính song song cho phép giảm thời gian thi hành các truy vấn và gia tăng lượng giao tác được thực hiện. Hiện nay, có các kiến trúc máy song song được sử dụng: kiến trúc mọi thứ dùng chung, kiến trúc dùng chung đĩa, kiến trúc không chia sẻ, kiến trúc phân cấp là sự pha trộn giữa các kiến trúc trên. Nhằm giải quyết vấn đề bế tắc vào ra thường gặp trong các hệ CSDL song song, ngoài việc áp dụng một kiến trúc phần cứng thích hợp thì chúng ta phải tiến hành phân hoạch dữ liệu một cách hợp lý cho các bộ xử lý, các Hệ quản trị CSDL trên các máy nhiều bộ xử lý thường sử dụng các kỹ thuật phân hoạch dữ liệu: phân hoạch theo vòng tròn Robin, phân hoạch theo hàm băm, phân hoạch theo khoảng, phân hoạch theo mảng nhiều chiều. Việc khai thác tiềm năng xử lý song song có thể được thực hiện theo các cơ chế khác nhau: song song liên truy vấn, song song nội truy vấn, song song nội toán tử. Song song liên truy vấn là thực hiện nhiều truy vấn cùng một lúc bằng cách lập lịch thực hiện cho các toán tử của các truy vấn đó. Điều này cho phép gia tăng thông lượng hệ thống. Song song nội truy vấn là dạng song song hoá thi hành song song một truy vấn đơn trên nhiều bộ xử lý và đĩa. Nghĩa là, nó thực hiện từng truy vấn một và cho phép thực hiện đồng thời các phép toán trên truy vấn đó. Song song một truy vấn cho phép gia tăng tốc độ thực hiện truy vấn đó. Song song nội toán tử là thực hiện một phép toán quan hệ bằng cách dùng nhiều bộ xử lý.

Một vấn đề mà tất cả các Hệ quản trị CSDL cho phép các truy vấn khai báo phải giải quyết là bài toán Tối ưu hoá truy vấn. Trên môi trường song song, một mô hình tối ưu hoá truy vấn dựa trên

cách tiếp cận hai giai đoạn do W. Hong đề xuất đã được nhiều nhà tin học quan tâm phát triển. Luận văn này đã dành phần lớn trình bày để khảo sát giai đoạn đầu của cách tiếp cận này nhằm: cực tiểu hoá khối lượng truy vấn. Các thuật toán được trình bày, dựa trên kỹ thuật quy hoạch động, có tính đến các chi phí phân bố lại, là một đóng góp bổ sung cho giai đoạn tối ưu hoá truy vấn. Thuật toán Tách Màu cho phép xác định thuộc tính phân bố tại từng nút của cây truy vấn. Thuật toán Tách Màu Mở Rộng cho phép xác định các chiến lược thực hiện tối ưu cho từng phép toán trên cây. Thuật toán Thứ Tự Phép Kết Vật Lý cho phép xác định thứ tự các phép kết nối tốt nhất cho một câu truy vấn *kết nối-chọn-chiều*. Các thuật toán này có thể có hai cách sử dụng:

- Kết hợp với bộ tối ưu hoá quy ước cho Bộ tối ưu hoá trong các máy song song.
- Tích hợp cả 3 thuật toán để thay thế cho bộ Tối ưu hoá qui ước.

Để minh hoạ cho các thuật toán trên, luận văn trình bày một cài đặt trên Visual Basic. Các cài đặt chủ yếu để chứng minh tính khả thi của thuật toán.

2. Hướng phát triển

Hướng phát triển của luận văn: tìm hiểu về vấn đề “lập lịch tối ưu cho cây truy vấn song song”. Nghĩa là, chúng ta sẽ tiếp cận bài toán lập lịch tối ưu cho cây toán tử nhận được từ giai đoạn JOQR, nhằm tìm kiếm một sự phân chia hợp lý các nút của cây toán tử cho các bộ xử lý để thời gian trả lời truy vấn là nhỏ nhất.