

Nội dung

- Thiết kế CSDL phân tán và song song
 - Phân mảnh
 - Phân tán dữ liệu
 - Các phương pháp kết hợp

71

Cấp phát (/định vị) mảnh

■ Đặt bài toán

Cho

$F = \{F_1, F_2, \dots, F_n\}$ các mảnh
 $S = \{S_1, S_2, \dots, S_m\}$ các trạm (vị trí) trên mạng
 $Q = \{q_1, q_2, \dots, q_d\}$ các ứng dụng

Tìm cách phân tán "tối ưu" F đến S .

■ Tối ưu

- Chi phí thấp nhất
 - Truyền thông + lưu trữ + xử lý (đọc & cập nhật)
 - Chi phí về thời gian (thông thường)
- Hiệu năng
 - Thời gian đáp ứng và/hoặc thông lượng
- Ràng buộc
 - Các ràng buộc trên mỗi trạm (lưu trữ & xử lý)

72

Thông tin yêu cầu

- Thông tin về CSDL
 - Chọn các mảnh
 - Kích thước mảnh
- Thông tin về ứng dụng
 - Các loại truy nhập và số lượng
 - Vị trí truy nhập
- Thông tin về mạng truyền thông
 - Bảng thông
 - Độ trễ
 - Chi phí truyền thông
- Thông tin về hệ thống máy tính
 - Chi phí đơn vị lưu trữ dữ liệu tại trạm (vị trí)
 - Chi phí đơn vị xử lý dữ liệu tại trạm

73

Cấp phát

Cấp phát file (FAP) và Cấp phát CSDL (DAP):

- Các mảnh không phải là các file riêng lẻ
 - Mỗi quan hệ cần phải được duy trì
- Truy nhập CSDL phức tạp hơn
 - Mô hình truy nhập file từ xa không áp dụng được
 - Mỗi quan hệ giữa cấp phát và xử lý truy vấn
- Cần xem xét chi phí thực thi tính nhất quán
- Cần xem xét chi phí điều khiển đồng thời

74

Mô hình cấp phát

Công thức tổng quát

min(Tổng chi phí)
tùy thuộc vào
ràng buộc thời gian đáp ứng
ràng buộc lưu trữ
ràng buộc xử lý

Quyết định

$$x_{ij} = \begin{cases} 1 & \text{nếu mảnh } F_i \text{ được lưu trữ tại trạm } S_j \\ 0 & \text{ngược lại} \end{cases}$$

75

Mô hình cấp phát

■ Tổng chi phí

$$\sum_{\text{tất cả các truy vấn}} \text{chi phí xử lý truy vấn} + \sum_{\text{tất cả các trạm}} \sum_{\text{tất cả các mảnh}} \text{chi phí lưu một mảnh tại một trạm}$$

■ Chi phí lưu trữ (của mảnh F_j tại S_k)

$$(\text{chi phí lưu trữ đơn vị tại } S_k) * (\text{kích thước của } F_j) * x_{jk}$$

■ Chi phí xử lý truy vấn (cho từng truy vấn)

thành phần xử lý + thành phần truyền dẫn

76

Mô hình cấp phát

■ Chi phí xử lý truy vấn

Thành phần xử lý

Chi phí truy nhập + chi phí thực thi tính toán vẹn + chi phí điều khiển đồng thời

□ Chi phí truy nhập

$$\sum_{\text{tất cả các trạm}} \sum_{\text{tất cả các mảnh}} (\text{số lượng truy nhập cập nhật} + \text{số lượng truy nhập đọc}) * x_{ij} * \text{chi phí xử lý cục bộ tại trạm}$$

□ Chi phí thực thi tính toán vẹn và kiểm soát đồng thời

- Có thể tính tương tự

77

Mô hình cấp phát

■ Chi phí xử lý truy vấn

Thành phần truyền dẫn

chi phí xử lý cập nhật + chi phí xử lý truy xuất

□ Chi phí cập nhật

$$\sum_{\text{tất cả các trạm}} \sum_{\text{tất cả các mảnh}} \text{Chi phí thông điệp cập nhật} + \sum_{\text{tất cả các trạm}} \sum_{\text{tất cả các mảnh}} \text{Chi phí báo nhận}$$

□ Chi phí truy xuất

$$\sum_{\text{tất cả các mảnh}} \min_{\text{tất cả các trạm}} (\text{chi phí lệnh truy xuất} + \text{Chi phí gửi phản hồi lại kết quả})$$

78

Mô hình cấp phát

■ Các ràng buộc

□ Thời gian đáp ứng

Thời gian thực thi truy vấn \leq thời gian phản hồi cho phép tối đa cho truy vấn đó

□ Ràng buộc lưu trữ (cho một trạm)

\sum yêu cầu lưu trữ của một mảnh tại trạm đó \leq khả năng lưu trữ tại trạm đó
Tất cả các mảnh

□ Ràng buộc xử lý (cho một trạm)

\sum Tài xử lý truy vấn tại trạm đó \leq khả năng xử lý tại trạm đó
Tất cả các truy vấn

79

Mô hình cấp phát

■ Phương pháp giải quyết

□ FAP là NP-đầy đủ

□ DAP cũng là NP-đầy đủ

■ Dựa trên các phương pháp Heuristics

□ Vị trí kho hàng hóa duy nhất (cho FAP) \rightarrow dễ tiếp cận

□ Bài toán xếp đồ vào ba lô

□ Các kỹ thuật nhánh cận (/nhánh và giới hạn)

□ Luồng mạng

80

Mô hình cấp phát

■ Một số giả thiết giúp giảm bớt không gian giải pháp

□ Giả sử tất cả các phân vùng ứng viên đã biết; chọn phân vùng "tốt nhất"

□ Bỏ qua nhân bản ban đầu

□ Cửa sổ trượt trên các mảnh

81

Nội dung

■ Thiết kế CSDL phân tán và song song

□ Phân mảnh

□ Phân tán dữ liệu

□ Các phương pháp kết hợp

82

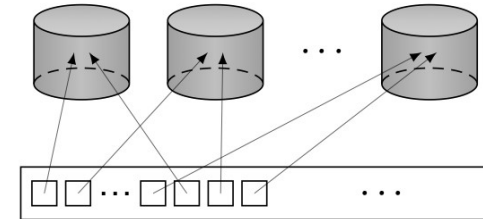
Kết hợp phân mảnh & Cấp phát

Phân vùng dữ liệu để xác định vị trí sẽ đặt nó.

- Các kỹ thuật được sử dụng khi không biết chắc chắn khối lượng công việc (Workload-agnostic techniques)
 - Phân vùng vòng tròn (Round-robin)
 - Phân vùng băm (Hash)
 - Phân vùng phạm vi (Range)
- Các kỹ thuật được sử dụng khi nhận biết được khối lượng công việc
 - Cách tiếp cận dựa trên đồ họa

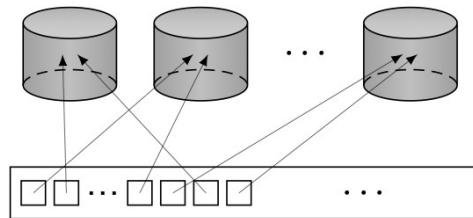
83

Phân vùng vòng tròn (Round-robin)



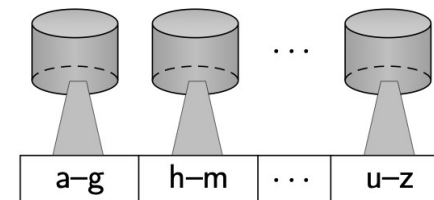
84

Phân vùng băm (Hash)



85

Phân vùng phạm vi (Range)

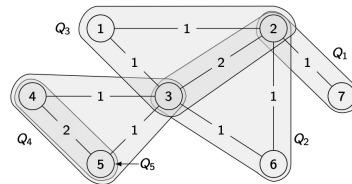


86

Phân vùng nhận biết khối lượng công việc

■ Ví dụ: **Schism**

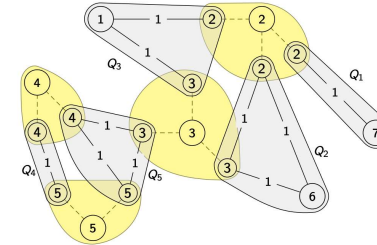
- Đồ thị $G=(V,E)$ trong đó
 - Đỉnh $v_i \in V$ biểu diễn một bộ trong CSDL,
 - Cạnh $e=(v_i, v_j) \in E$ biểu diễn một truy vấn truy nhập đến cả hai bộ v_i và v_j ;
 - Mỗi cạnh có một trọng số đếm số lượng truy vấn truy nhập đến cả hai bộ
- Thực hiện phân vùng đồ thị rời rạc đỉnh
 - Mỗi đỉnh được gán cho một vùng riêng biệt



87

Kết hợp nhân bản

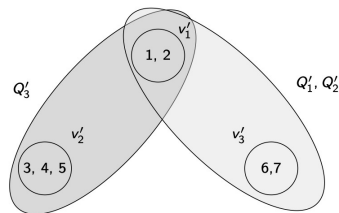
- Nhân bản mỗi đỉnh dựa trên số lượng giao dịch truy nhập vào bộ đó → mỗi giao dịch truy nhập vào một bản sao riêng.



88

Xử lý kích thước đồ thị

- Mỗi bộ một đỉnh → đồ thị quá lớn → thư mục quá lớn
- **SWORD**
 - Sử dụng mô hình siêu đồ thị
 - Nén thư mục



89

Các cách tiếp cận thích ứng

- Thiết kế lại khi xảy ra các thay đổi **vật lý** (đặc điểm mạng, dung lượng lưu trữ khả dụng) và **logic** (khối lượng công việc).
- Hầu hết tập trung vào logic
- Hầu hết theo cách tiếp cận kết hợp
- Ba vấn đề:
 - ❶ Làm thế nào để phát hiện những thay đổi về khối lượng công việc?
 - ❷ Làm thế nào để xác định các mục dữ liệu bị ảnh hưởng?
 - ❸ Làm thế nào để thực hiện thay đổi một cách hiệu quả?

90

Phát hiện thay đổi khối lượng công việc

- Không có nhiều việc
- Phân tích nhật ký hệ thống định kỳ
- Giám sát liên tục khối lượng công việc trong hệ quản trị CSDL
 - SWORD: số lượng truy vấn phân tán
 - E-Store: giám sát các số liệu cấp hệ thống (ví dụ, mức sử dụng CPU) và truy nhập mức bộ (dữ liệu).

91

Phát hiện các mục dữ liệu bị ảnh hưởng

- Phụ thuộc vào phương pháp phát hiện thay đổi khối lượng công việc
- Nếu giám sát truy vấn → các truy vấn sẽ xác định các mục dữ liệu
 - Apollo: khái quát hóa từ các truy vấn "tương tự"


```
SELECT PNAME FROM PROJ WHERE BUDGET>20000 AND LOC='LONDON'
```

Sẽ được khái quát hóa thành:

```
SELECT PNAME FROM PROJ WHERE BUDGET?? AND LOC='?'
```
- Nếu giám sát truy nhập mức bộ dữ liệu (E-Store), thì cũng sẽ biết được.

92

Thực hiện thay đổi

- Định kỳ tính toán phân tán lại
 - Không hiệu quả
- Tính toán gia tăng và di chuyển
 - Biểu diễn đồ thị → quan sát những thay đổi trong đồ thị
 - SWORD và AdaptCache: phân vùng đồ thị gia tăng bắt đầu di chuyển dữ liệu để cấu hình lại
 - E-Store: xác định các bộ dữ liệu chuẩn bị được di chuyển theo kế hoạch; cũng như cấp phát lại các bộ
 - Vấn đề tối ưu hóa; các giải pháp heuristic thời gian thực
 - Bề khóa CSDL: liên tục sắp xếp lại dữ liệu để phù hợp với khối lượng công việc truy vấn
 - Các truy vấn đến được sử dụng như là khuyến nghị
 - Khi một nút cần dữ liệu cho một truy vấn cục bộ, điều này cho thấy là dữ liệu có thể cần phải được di chuyển

93

Tài liệu tham khảo

- M. Tamer Özsu , Patrick Valduriez, "Principles of Distributed Database Systems", Fourth Edition, Springer, 2020.
<https://link.springer.com/book/10.1007/978-3-030-26253-2>

94