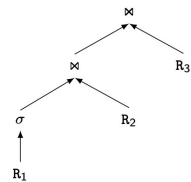


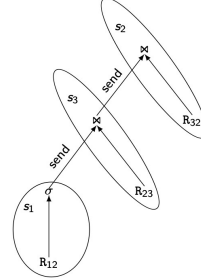
## Tối ưu hóa 2-bước

1. Tại thời điểm biên dịch, tạo một kế hoạch tĩnh chỉ với thứ tự thao tác và các phương thức truy nhập
2. Tại thời điểm khởi động, thực hiện lựa chọn trạm và sao chép lựa chọn cũng như định vị các hoạt động cho các trạm.

Kế hoạch tĩnh



Kế hoạch thời gian chạy



63

## Định nghĩa bài toán 2-bước

### Cho

- Tập các trạm  $S = \{s_1, s_2, \dots, s_n\}$  với tải trọng tại mỗi trạm.
- Một truy vấn  $Q = \{q_1, q_2, q_3, q_4\}$  sao cho mỗi truy vấn con  $q_i$  là đơn vị xử lý tối đa truy nhập vào một quan hệ và truyền thông với các truy vấn hàng xóm của nó.
- Với mỗi  $q_i$  trong  $Q$ , một tập các trạm định vị khả thi  $S_{q_i} = \{s_1, s_2, \dots, s_k\}$  trong đó mỗi trạm lưu trữ một bản sao của quan hệ trong  $q_i$ .

### Mục tiêu là tìm ra phân bổ tối ưu của $Q$ cho $S$ sao cho

- Sự mất cân bằng tải của  $S$  được giảm thiểu
- Tổng chi phí truyền thông được giảm thiểu

64

## Thuật toán 2-bước

- Với mỗi  $q$  trong  $Q$  tính toán tải ( $S_q$ )
- Chùng nào  $Q$  không rỗng thì thực hiện:
  1. Chọn truy vấn con  $a$  với độ linh hoạt phân bổ ít nhất
  2. Chọn trạm tốt nhất  $b$  cho  $a$  (với tải ít nhất và lợi ích tốt nhất)
  3. Loại bỏ  $a$  khỏi  $Q$  và tính toán lại tải nếu cần

65

## Ví dụ thuật toán 2-bước

- Đặt  $Q = \{q_1, q_2, q_3, q_4\}$  trong đó  $q_1$  liên kết với  $R_1$ ,  $q_2$  liên kết với  $R_2$  được kết nối với kết quả  $q_1, \dots$
- Lặp 1: chọn  $q_4$ , phân bổ cho  $s_1$ , đặt tải ( $s_1$ )=2
- Lặp 2: chọn  $q_2$ , phân bổ tới  $s_2$ , đặt tải ( $s_2$ )=3
- Lặp 3: chọn  $q_3$ , phân bổ tới  $s_1$ , đặt tải ( $s_1$ )=3
- Lặp 4: chọn  $q_1$ , phân bổ tới  $s_3$  hoặc  $s_4$

Sites	Load	$R_1$	$R_2$	$R_3$	$R_4$
$s_1$	1	$R_{11}$		$R_{31}$	$R_{41}$
$s_2$	2		$R_{22}$		
$s_3$	2	$R_{13}$		$R_{33}$	
$s_4$	2	$R_{14}$	$R_{24}$		

**Chú ý:** nếu trong lần lặp 2,  $q_2$  đã được phân bổ tới  $s_4$ , thì sẽ tạo ra một kế hoạch tốt hơn. Như vậy, việc tối ưu hóa lại vẫn có thể bỏ sót các phương án tối ưu.

## Nội dung

- Xử lý truy vấn phân tán
  - Phân rã và cục bộ hóa truy vấn
  - Tối ưu hóa truy vấn phân tán
  - Trình tự kết nối
  - Xử lý truy vấn thích ứng

67

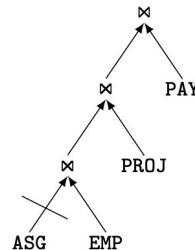
## Xử lý truy vấn thích ứng – Động lực

- Giả thiết tối ưu hóa truy vấn cơ bản
  - Bộ tối ưu hóa có đủ thông tin về thời gian chạy
    - Thông tin chi phí
  - Điều kiện thời gian chạy vẫn ổn định trong quá trình thực hiện truy vấn
- Thích hợp cho các hệ thống có ít nguồn dữ liệu trong môi trường được kiểm soát
- Không phù hợp trong môi trường thay đổi với số lượng lớn nguồn dữ liệu và điều kiện thời gian chạy không thể đoán trước

68

## Ví dụ: Kế hoạch thực thi truy vấn (QEP) với toán tử bị chặn

- Giả sử mỗi quan hệ ASG, EMP, PROJ và PAY được đặt tại trạm khác nhau.
- Nếu trạm ASG ngừng hoạt động, thì toàn bộ đường kết nối bị chặn
- Tuy nhiên, với một vài tổ chức lại, kết nối EMP và PAY vẫn có thể được thực hiện trong khi chờ ASG



69

## Xử lý truy vấn thích ứng – Định nghĩa

- Quá trình xử lý truy vấn có tính thích ứng nếu nó nhận thông tin từ môi trường thực thi và xác định hành vi của nó tương ứng
  - Vòng lặp phản hồi giữa bộ tối ưu hóa và môi trường thời gian chạy
  - Truyền thông tin thời gian chạy giữa các thành phần của hệ quản trị CSDLPT
- Các thành phần bổ sung
  - Theo dõi, đánh giá, phản ứng
  - Được nhúng trong các toán tử điều khiển của QEP
- Cân bằng giữa khả năng phản ứng và chi phí thích ứng

70

## Các thành phần thích ứng

- Các tham số giám sát (được thu thập bởi các cảm biến trong QEP)
  - Kích thước bộ nhớ
  - Tỷ lệ dữ liệu đến
  - Thống kê thực tế
  - Chi phí thực thi toán tử
  - Thông lượng mạng
- Phản ứng thích ứng
  - Thay đổi lịch trình
  - Thay thế toán tử bằng một toán tử tương đương
  - Sửa đổi hành vi của một toán tử
  - Phân vùng lại dữ liệu

71

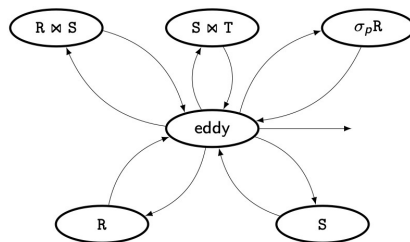
## Cách tiếp cận xoáy (Eddy)

- Biên dịch truy vấn: tạo ra một bộ  $\langle D, P, C, \text{Eddy} \rangle$ 
  - $D$ : tập các nguồn dữ liệu (ví dụ, các quan hệ)
  - $P$ : tập các vị từ
  - $C$ : các ràng buộc về thứ tự phải được tuân theo trong thời gian chạy
  - Toán tử Eddy: toán tử  $n$ -ary giữa  $D$  và  $P$
- Thực thi truy vấn: toán tử sắp xếp dựa trên cơ sở bộ dữ liệu sử dụng Eddy
  - Định tuyến bộ nhanh chóng tới các toán tử dựa trên chi phí và phép chọn.
  - Thay đổi thứ tự phép kết nối trong quá trình thực thi
    - Cần các thuật toán kết nối đối xứng như, ví dụ Ripple

72

## QEP với Eddy

- $D = \{R, S, T\}$
- $P = \{\sigma_P(R), R \bowtie_1 S, S \bowtie_2 T\}$
- $C = \{S < T\}$  trong đó  $<$  đặt các bộ  $S$  thăm dò các bộ  $T$  bằng cách sử dụng chỉ mục trên thuộc tính nối
  - Truy nhập tới  $T$  được bao bọc bởi  $\bowtie$



73

## 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>

74