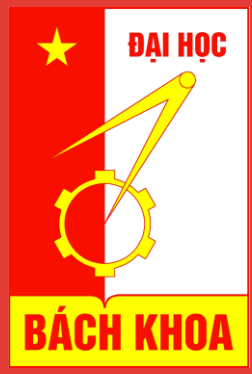

Quản trị cơ sở dữ liệu và Tối ưu hiệu năng

Tối ưu tương tranh (concurrency tuning)

Danh mục

- Tối ưu tương tranh
  - Chia giao dịch (Transaction Chopping)

Phân chia giao dịch dài (Chopping Long Transactions)

- Các giao dịch ngắn hơn
  - Yêu cầu khóa ít hơn (do đó các giao dịch này ít bị chặn hơn hoặc chặn một giao dịch khác)
  - Yêu cầu các giao dịch khác đợi khóa ít hơn
  - Tốt hơn cho việc ghi file log (logging)
- Phân chia giao dịch (Transaction chopping):
  - Chia các giao dịch dài thành các giao dịch ngắn
  - Không phá vỡ tính đúng đắn

Thuật ngữ

- Giao dịch (Transaction): là một chuỗi các truy cập đĩa cho việc đọc/ghi
- Thành phần của giao dịch (Piece of transaction): Là chuỗi con liên tiếp của hoạt động truy cập dữ liệu.
  - Ví dụ giao dịch T: R (A), R (B), W (A)
  - R (A) và R (A), R (B) là các thành phần của T
  - R (A), W (A) không là thành phần của T (vì không liên tục)
- Phân chia (Chopping): Chia giao dịch thành các thành phần.
  - Ví dụ giao dịch T: R (A), R (B), W (A)
  - T1:R (A), R (B) and T2 : W (A) là một phân chia của T

Chia giao dịch dài – Ví dụ 1

- Ngân hàng với các tài khoản và các chi nhánh
  - Mỗi tài khoản được ấn định đến một chi nhánh một cách chính xác
  - Số dư chi nhánh là tổng của tất cả các tài khoản trong chi nhánh đó
  - Khách hàng có thể rút tiền mặt trong ngày
- Các giao dịch qua đêm:
  - Cập nhật giao dịch: Phản ánh việc rút tiền hàng ngày trong CSDL
  - Kiểm tra số dư tài khoản: Khách hàng yêu cầu kiểm tra số dư tài khoản (chỉ đọc)
- Giao dịch cập nhật T_{blob}
  - Cập nhật tất cả số dư tài khoản để phản ánh việc rút tiền hàng ngày
  - Cập nhật các số dư chi nhánh tương ứng
- Vấn đề: Các hoạt động kiểm tra số dư tài khoản bị chặn bởi T_{blob} và mất quá nhiều thời gian

Chia các giao dịch dài – Ví dụ 1

- Chia các giao dịch cập nhật T_{blob} thành nhiều giao dịch nhỏ
- Phương án 1: Mỗi hoạt động cập nhật tài khoản là một giao dịch:
  - Cập nhật một tài khoản
  - Cập nhật số dư chi nhánh tương ứng
- Phương án 2: Mỗi hoạt động cập nhật tài khoản bao gồm 2 giao dịch:
  - T_1 : cập nhật tài khoản
  - T_2 : Cập nhật số dư chi nhánh
- Lưu ý: Tính tách biệt không bao hàm cả tính nhất quán
  - Cả 2 phương án đều duy trì tính tuần tự (tách biệt)
  - Phương án 2: Tính nhất quán (Tổng các tài khoản bằng với số dư chi nhánh) được chấp nhận chỉ khi một trong T1 hoặc T2 thực hiện cam kết (commit)

Chia các giao dịch dài – Ví dụ 2

- Kịch bản về ngân hàng giống với ví dụ 1.
- Các giao dịch:
  - Giao dịch cập nhật: Mỗi giao dịch cập nhật một tài khoản và số dư ngân hàng tương ứng (Phương án 1 trong ví dụ 1)
  - Kiểm tra số dư tài khoản: Khách hàng yêu cầu kiểm tra số dư tài khoản (Chỉ đọc)
  - Tính nhất quán (T'): Tính tổng số dư tài khoản cho mỗi chi nhánh và so sánh với số dư chi nhánh
- Chia: T' có thể chia thành các giao dịch cho mỗi chi nhánh riêng biệt.
- Tính tuần tự được duy trì:
  - Hoạt động kiểm tra tính nhất quán trên các chi nhánh khác nhau không chia sẻ mục dữ liệu
  - Việc cập nhật giúp CSDL trong trạng thái nhất quán cho T'
- Lưu ý: Giao dịch cập nhật không thể chia nhỏ hơn được (Phương án 2)!
- Bài học kinh nghiệm:
  - Đôi lúc các giao dịch có thể được chia mà không phải hi sinh tính tuần tự
  - Thêm một giao dịch mới vào việc cài đặt có thể làm mất hiệu lực tất cả các phân chia (chopping) trước đó

Hình thức tiếp cận phân chia (Formal Chopping Approach)

- Giả thuyết: Khi nào phân chia được áp dụng?
- Quy tắc thực hiện: Các giao dịch đã được phân chia phải thực hiện như thế nào?
- Đồ thị phân chia (Chopping graph): Các chia nào là chính xác?

Giả thiết cho phân chia giao dịch (Assumptions for Transaction Chopping)

- Giao dịch: Tất cả các giao dịch thực hiện trong một khoản đã biết
- Hủy giao dịch (Rollbacks): Nó được biết nơi mà việc hủy giao dịch được gọi
- Thất bại: Trong trường hợp thất bại, nó có thể xác định các giao dịch nào đã hoàn thành và không hoàn thành
- Biến: Mã giao dịch thay đổi một biến chương trình x phải dùng lại được (reentrant), ví dụ: Nếu giao dịch bị hủy bỏ do xung đột đồng thời và việc thực hiện một cách chính xác, x được chuyển đến một trạng thái nhất quán

Luật thực hiện (Execution Rules)

- Thứ tự thực hiện: Việc thực hiện của các thành phần phải tuân theo thứ tự của giao dịch
- Xung đột khóa: Nếu một thành phần bị hủy bỏ do xung đột khóa, sau đó nó sẽ đệ trình lại (resubmitted) cho đến khi thực hiện cam kết
- Hủy giao dịch: Nếu một thành phần bị hủy bỏ do việc hủy giao dịch, khi đó các thành phần khác sẽ không được thực hiện

Bài toán phân chia giao dịch (The Transaction Chopping Problem)

- Cho: Tập A = {T_1, T_2, …, T_n} của (có thể) các giao dịch đồng thời
- Mục tiêu: Tìm một phân chia B của các giao dịch trong A sao cho bất kì việc thực hiện tuần tự nào của các giao dịch trong B (Theo các luật thực hiện) tương đương với một số việc thực hiện nối tiếp của giao dịch trong A.

  Các chia như vậy được gọi là chính xác.
- Lưu ý: Việc thực hiện "tuần tự" (serializable) của B có thể đồng thời, theo một giao thức cho tính tuần tự

Đồ thị phân chia (Chopping Graph)

- Chúng ta thể hiện một phân chia cụ thể của các giao dịch bằng một đồ thị
- Đồ thị phân chia: Là đồ thị vô hướng với 2 loại cạnh:
  - Nút: Mỗi thành phần trong phân chia là là một nút
  - Cạnh-C: Cạnh giữa 2 thành phần xung đột bất kì
  - Cạnh-S: Cạnh giữa 2 thành phần anh chị em bất kì
- Thành phần xung đột: 2 thành phần p và p’ xung đột nếu:
  - p và p’ là các thành phần của các giao dịch ban đầu khác nhau
  - Cả p và p’ cùng truy cập vào một mục dữ liệu x và ít nhất một thành phần làm thay đổi nó
- Thành phần anh chị em: 2 thành phần p và p’ là anh chị em nếu:
  - p và p’ là các thành phần lân cận của cùng một giao dịch ban đầu

Đồ thị phân chia – ví dụ

- Ghi chú: phân chia của các giao dịch đồng thời có thể.
  - Các giao dịch ban đầu được kí hiệu: T_1, T_2, …
  - Phân chia T_i thành các phần : T_{i1} , T_{i2} ,…
- Ví dụ các giao dịch: (T1 : R (x), R (y), W (y) được chia thành T11, T12)
  - T11 : R (x)
  - T12 : R (y), W (y)
  - T2 : R (x), W (x)
  - T3 : R (y), W (y)
- Cạnh xung đột giữa các nút:
  - T11 và T2 (Xung đột trên x)
  - T12 và T3 (Xung đột trên y)
- Cạnh anh chị em giữa các nút:
  - T11 và T22 (cùng một giao dịch ban đầu T1)

Chống Hủy giao dịch (Rollback Safe)

- Động cơ: Giao dịch T được chia thành T_1 và T_2.
  - T_1 thực hiện và cam kết
  - T_2 chứa một lệnh hủy giao dịch và phục hồi (roll back)
  - T_1 đã cam kết và sẽ không phục hồi
  - Việc hủy giao dịch trong giao dịch ban đầu T cũng có thể khôi phục lại kết quả của thành phần T_1 !
- Một phân chia của giao dịch T gọi là chống hủy giao dịch (rollback save) nếu:
  - T không có lệnh hủy giao dịch hoặc
  - Tất cả các lệnh hủy giao dịch nằm trong thành phần đầu tiên của giao dịch

Phân chia chính xác (Correct Chopping)

Định lý (phân chia chính xác):

Một phân chia là chính xác nếu nó là chống hủy giao dịch (rollback save) và đồ thị của phân chia đó không chứa chu kỳ-SC (SC-cycles)

- Phân chia của các ví dụ trước là chính xác (không có chu kì-SC và hủy giao dịch)
- Nếu một phân chia là không chính xác, khi đó bất kì phân chia khác của bất kì giao dịch nào sẽ không trả về một phân chia chính xác.
- Nếu 2 thành phần của giao dịch T ở trong một chu kì-SC như một kết quả của phân chia T, khi đó chúng sẽ ở trong một chu kì ngay cả khi không có những giao dịch khác (khác T) bị chia.

Phân chia riêng (Private Chopping)

- Phân chia riêng (Private chopping): Cho các giao dịch: T_1 , T_2 ,…,T_n . T_{i1} , T_{i2} , …,T_{ik} là một phân chia riêng của T_i nếu :
  - Không có chu kì-SC trong đồ thị với các nốt sau: $\{T_1,\ldots, T_{i1} ,\ldots, T_{ik} ,\ldots, T_n\}$
  - T_i là chống hủy giao dịch
- Luật phân chia riêng: Phân chia bao gồm private(T_1), private(T_2),…, private(T_n) là chính xác
- Hệ quả:
  - Mỗi giao dịch T_i có thể được chia một cách tách bạch, dẫn đến private(T_i)
  - Toàn bộ phân chia là sự kết hợp của các phân chia riêng

Thuật toán phân chia (Chopping Algorithm)

- Vẽ một cạnh-S giữa các thao tác R/W của một giao dịch duy nhất.
- Với mỗi mục dữ liệu x cung cấp một danh sách ghi, ví dụ: một danh sách của các giao dịch để ghi mục dữ liệu đó.
- Với mỗi R(x) hoặc W(x) trong tất cả các giao dịch:
  - Tìm các giao dịch xung đột trong danh sách ghi của x
  - Vẽ một cạnh-C đến các thao tác xung đột tương ứng
- Loại bỏ tất cả các cạnh-S liên quan đến một chu kì-SC.

Thuật toán phân chia – Ví dụ

- Các giao dịch: (Rx = R (x), Wx = W (x))
  - T1 : Rx , Wx, Ry , Wy
  - T2 : Rx , Wx
  - T3 : Ry , Rz , Wy
- Các danh sách ghi: x:T_1 ,T_2 ; y:T_1 , T_3 ; z: Ø
- Các cạnh-C:
  - T1: Rx − T2.Wx, Wx − T2.Wx, Ry − T3.Wy , Wy − T3.Wy
  - T2: Rx − T1.Wx (Wx − T1.Wx: đã có trong T1)
  - T3: Ry − T1.Wy (Wy − T1.Wy : đã có trong T1)
- Loại bỏ các cạnh-S: T1: Rx − Wx, Ry − Wy ; T2: Rx − Wx; T3: Ry − Rz , Rz − Wy
- Phân chia cuối cùng:
  - T11 : Rx , Wx; T12 : Ry , Wy
  - T2 : Rx , Wx
  - T3 : Ry , Rz , Wy

Sắp xếp lại giao dịch (Reordering Transactions)

- Thao tác giao hoán:
  - Thay đổi thứ tự không thay đổi ngữ nghĩa của chương trình
  - Ví dụ: R (y), R (z), W (y ← y + z) và R (z), R (y), W (y ← y + z) thực hiện giống nhau
- Phân chia giao dịch:
  - Thay dổi thứ tự của các thao tác giao hoán có thể đem lại phân chia tốt hơn
  - Tránh nhiệm của lập trình viên là kiểm tra các thao tác đó có giao hoán hay không.
- Ví dụ: Xét T_3 : R_y , R_z , W_y của ví dụ trước:
  - Giả sử T_3 tính y+z và lưu trữ tổng trong y
  - Khi đó R_y và R_z là giao hoán và có thể được hoán đổi
  - T_3' : R_z , R_y , W_y có thể được chia: T'_{31} : R_z , T'_{32} : R_y , W_y