

Quản trị cơ sở dữ liệu và Tối ưu hiệu năng

Tối ưu tương tranh
(concurrency tuning)

Danh mục

- Tối ưu tương tranh
 - Giảm các bảo đảm tách bạch (Weaken Isolation Guarantees)

Các hiện tượng không mong muốn của các giao dịch đồng thời

- Đọc bẩn (Dirty read)
 - Các giao dịch đọc dữ liệu được ghi bởi các giao dịch không cam kết đồng thời (concurrent uncommitted transaction)
 - Vấn đề: Việc đọc có thể trả về kết quả không có trong CSDL vì giao dịch ghi đã bị hủy bỏ.
- Đọc không lặp lại (Non-repeatable read)
 - Các việc đọc khác nhau trên cùng một mục dữ liệu (item) bên trong một giao dịch duy nhất cho những kết quả khác nhau (gây ra bởi các giao dịch khác)
 - Ví dụ: Các giao dịch đồng thời T1: $x = R(A)$, $y = R(A)$, $z = y - x$ và T2: $W(A = 2 * A)$, sau đó z có thể nhận giá trị 0 hoặc giá trị đầu của A (Có thể là 0)
- Đọc ảo tưởng (Phantom read)
 - Lặp lại cùng một truy vấn chậm hơn trước đó trong giao dịch sẽ cho một tập khác nhau của các bộ kết quả
 - Các giao dịch khác nhau có thể chen các bộ (tuples) mới trong khi có một tìm kiếm (scan) đang thực hiện
 - Ví dụ: “Q: Lấy các tài khoản với số dư tài khoản > 1000” cho 2 bộ tại thời điểm đầu tiên, sau đó một tài khoản mới với số dư tài khoản > 1000 được chen bởi những giao dịch khác, tại thời điểm thứ 2 Q cho kết quả là 3 bộ.

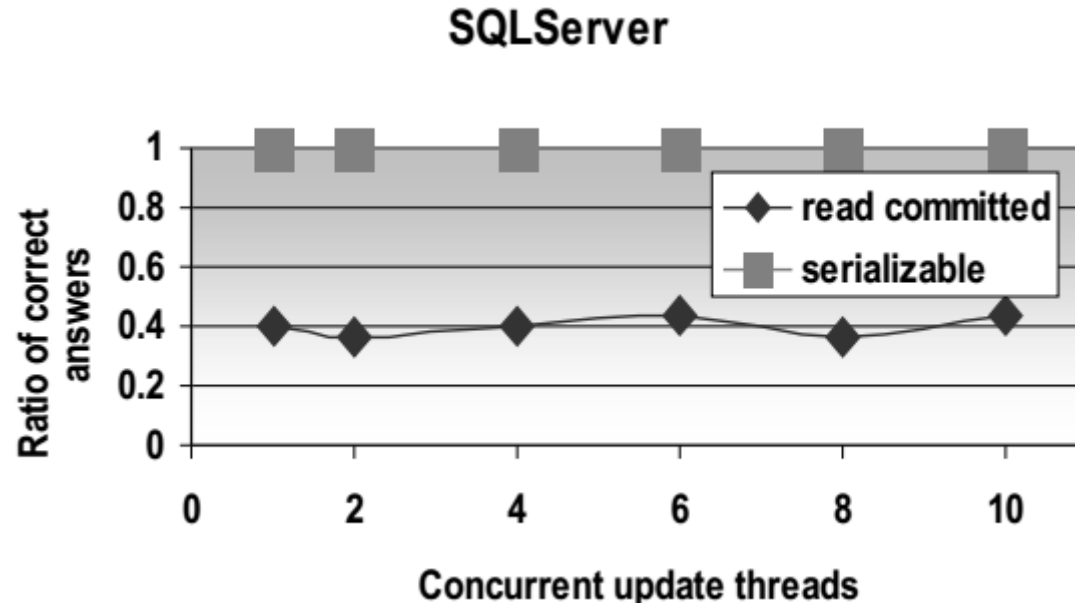
Các bảo đảm tách bạch (Chuẩn SQL) (Isolation Guarantees (SQL Standard))

- Đọc không cam kết (Read uncommitted): Đọc bẩn (dirty), đọc không lặp (non-repeatable), đọc ảo tưởng (phantom)
 - Các khóa đọc (read locks) được giải phóng sau khi đọc, các khóa viết (write locks) bị giảm cấp xuống khóa đọc sau khi viết, giảm cấp các khóa được giải phóng theo 2 giai đoạn khóa (2-phase locking)
 - Việc đọc có thể truy cập đến dữ liệu không cam kết (uncommitted data)
 - Việc ghi không ghi đè lên dữ liệu không cam kết
- Đọc cam kết (Read committed): Đọc không lặp, đọc ảo tưởng
 - Các khóa đọc được giải phóng sau khi đọc, các khóa viết dựa trên 2 giai đoạn khóa
 - Việc đọc chỉ có thể truy cập đến dữ liệu đã cam kết
 - Đại gặt bền (cursor stability): Ngoài ra, việc đọc có thể lặp lại bên trong một SELECT duy nhất
- Đọc lặp lại (Repeatable read): Đọc ảo tưởng
 - 2 giai đoạn khóa, nhưng không có các khóa miền (range locks)
 - Đọc ảo tưởng có thể xảy ra
- Tuần tự (Serializable):
 - Không một hiện tượng không mong muốn nào có thể xảy ra
 - Được thực hiện bởi 2 giai đoạn khóa với các khóa miền

Thực nghiệm: Đọc cam kết và tuần tự (Read Commit vs. Serializable)

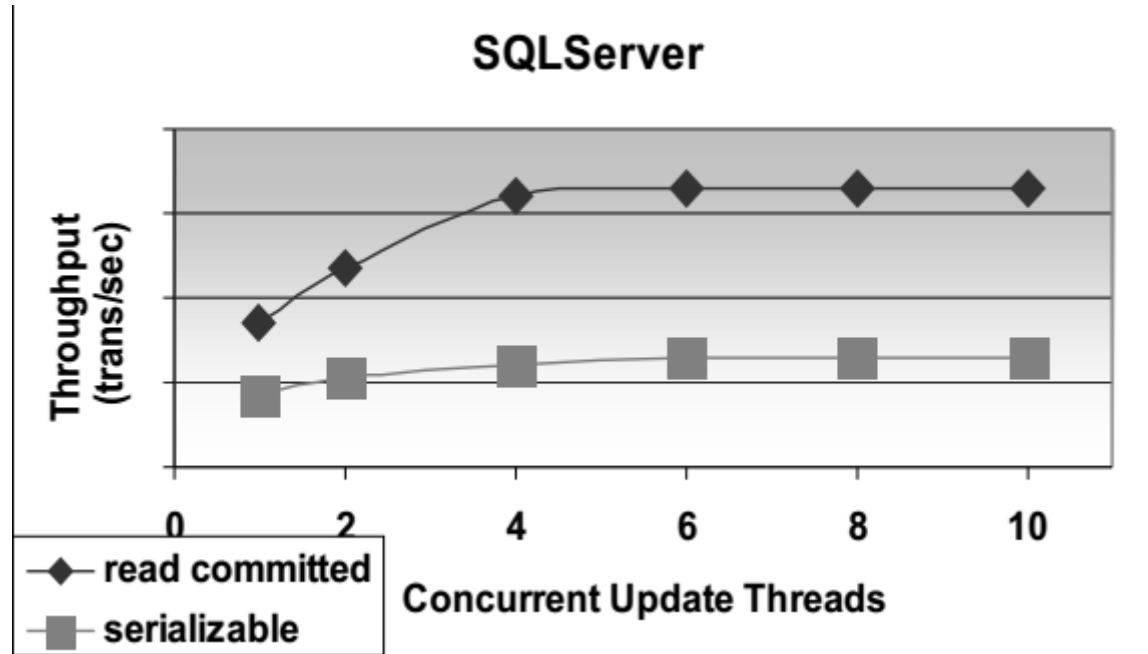
- Cài đặt thực nghiệm:
 - T1: Truy vấn lấy tổng: `SELECT SUM(balance) FROM Accounts`
 - T2: Chuyển tiền giữa các tài khoản
 - Khóa mức hàng
- Tham số: Số lượng các luồng đồng thời
- Đo:
 - Tỷ lệ các kết quả truy vấn đúng (Qua nhiều phép thử)
 - Đo thông lượng (throughput)

Thực nghiệm: Đọc cam kết và tuần tự (Read Commit vs. Serializable)



- Đọc cam kết công nhận tổng các số dư tài khoản sau khi họa động ghi nợ xảy ra, nhưng trước khi hoạt động cho vay được thực hiện – Tổng không đúng!

Thực nghiệm: Đọc cam kết và tuần tự (Read Commit vs. Serializable)



- Đọc cam kết: Nhanh hơn, nhưng những kết quả truy vấn không đúng
- Tuần tự: Luôn luôn đúng, nhưng thông lượng chậm hơn

Khi nào cần làm suy yếu các đảm bảo tách biệt

- Truy vấn không cần kết quả chính xác (ví dụ: truy vấn thống kê)
 - Ví dụ: Đếm tất cả các tài khoản có số dư $> \$1000$.
 - Đọc cam kết là đủ
- Các giao dịch với tương tác người:
 - Ví dụ: Hệ thống đặt trước chuyến bay
 - Chi phí của tuần tự (serializability) quá cao!

Thực nghiệm: Hệ thống đặt trước chuyến bay

- Việc đặt trước bao gồm 3 bước:
 1. Lấy danh sách các chỗ ngồi trống
 2. Để khách hàng lựa chọn
 3. Bảo đảm chỗ ngồi
- Giao dịch duy nhất:
 - Các chỗ ngồi bị khóa khi khách hàng lựa chọn
 - Tất cả các khách hàng khác bị chặn !
- 2 giao dịch: (1) lấy danh sách, (2) bảo đảm chỗ ngồi
 - Chỗ ngồi đã bị lấy khi khách hàng muốn đảm bảo chỗ ngồi đó
 - Nhiều sự cho phép hơn là chặn tất cả các khách hàng khác

Ảnh chụp tách bạch cho việc đọc dài – vấn đề

- Xét kịch bản cho một ngân hàng sau:
 - Truy vấn chỉ đọc Q: `SELECT SUM(deposit) FROM Accounts`
 - Cập nhật giao dịch T: Chuyển tiền giữa khách hàng A và B
- 2 giai đoạn khóa không hiệu quả cho các truy vấn chỉ đọc dài:
 - Các giao dịch chỉ đọc giữ khóa trên tất cả các mục dữ liệu đọc (read items)
 - Trong ví dụ của chúng tôi, T phải đợi Q kết thúc (Q chặn T)
 - Các bế tắc có thể xuất hiện:
 $T.xL(A), Q.sL(B), Q.sL(A) - \text{wait}, T.xL(B) - \text{wait}$
- Đọc cam kết có thể dẫn đến các kết quả không chính xác:
 - Trước khi giao dịch: $A = 50, B = 30$
 - Q : $sL(A), R(A) = 50, uL(A)$
 - T : $xL(A), xL(B), W(A \leftarrow A + 20), W(B \leftarrow B - 20), uL(A), uL(B)$
 - Q : $sL(B), R(B) = 10, uL(B)$
 - Tổng được tính bởi Q cho $A + B$ là 60 (thay vì 80)

Ảnh chụp tách bạch cho việc đọc dài (Snapshot Isolation for Long Reads)

- Ảnh chụp tách bạch: Các truy vấn chỉ đọc chính xác mà không có khóa
 - Các truy vấn chỉ đọc Q với ảnh chụp tách bạch
 - Nhớ các giá trị cũ của tất cả các mục dữ liệu bị thay đổi sau khi Q bắt đầu
 - Q thấy các giá trị của các mục dữ liệu khi Q bắt đầu
- Ví dụ: Kịch bản về ngân hàng với ảnh chụp tách bạch:
 - Trước khi giao dịch: $A = 50, B = 30$
 - Q : $sL(A), R(A) = 50, uL(A)$
 - T : $xL(A), xL(B), W(A \leftarrow A + 20), W(B \leftarrow B - 20), uL(A), uL(B)$
 - Q : $R(B) = 30$ (Đọc giá trị cũ)
 - Tổng được tính bởi Q cho $A + B$ là 80

Tính đồng thời trong Oracle (Concurrency in Oracle)

- “Đọc cam kết” (Read committed) trong Oracle có nghĩa là:
 - Đọc không lặp lại và đọc ảo tưởng có thể xảy ra tại mức giao dịch, nhưng không xảy ra bên trong một lệnh SQL đơn
 - Xung đột cập nhật: Nếu hàng đã được cập nhật, đợi cho cập nhật giao dịch để xác thực, sau đó cập nhật phiên bản hàng mới (Hoặc bỏ qua hàng nếu nó đã xóa) – không có hủy giao dịch (rollback)
 - Trạng thái mâu thuẫn có thể: Giao dịch thấy những cập nhật của những giao dịch khác chỉ trên các hàng cập nhật bởi chính nó
- “Tuần tự” (Serializable) trong Oracle có nghĩa:
 - Hiện tượng: Không có hiện tượng không mong muốn nào xảy ra
 - Xung đột cập nhật: Nếu 2 giao dịch cập nhật cùng một mục dữ liệu, giao dịch cập nhật mục dữ liệu đó sau phải hủy bỏ - Hủy giao dịch!
 - Không tuần tự: Ảnh chụp tách bạch không đảm bảo đầy đủ tính tuần tự (skew writes)
- Tương tự với PostgreSQL.

Skew Writes: Ảnh chụp tách bạch không tuần tự

- Ví dụ: $A = 3, B = 17$
 - T 1 : $A \leftarrow B$
 - T 2 : $B \leftarrow A$
- Thực hiện tuần tự:
 - Thứ tự: T 1, T 2: $A = B = 17$
 - Thứ tự: T 2, T 1: $A = B = 3$
- Ảnh chụp tách bạch:
 - T 1 : $R(B) = 17$
 - T 2 : $R(A) = 3$
 - T 1 : $W(A \leftarrow 17)$
 - T 2 : $W(B \leftarrow 3)$
 - Kết quả: $A = 17, B = 3$ (Khác biệt so với tuần tự)

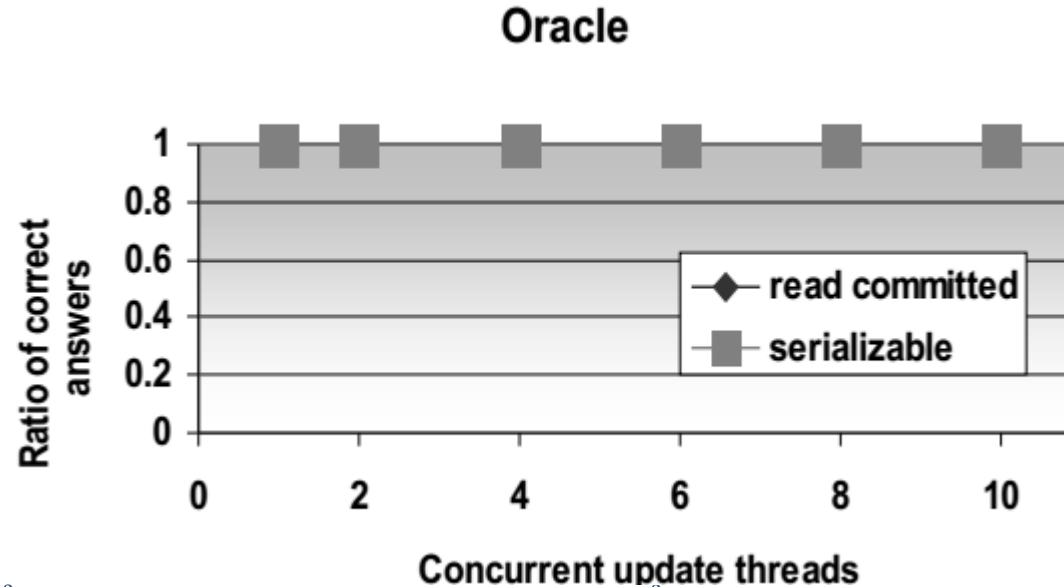
Ảnh chụp tách bạch

- Ưu điểm: (Giả sử “Tuần tự” của Oracle)
 - Các bộ đọc không chặn các bộ ghi (Giống với khóa)
 - Các bộ viết không khóa bộ đọc (Giống với khóa)
 - Bộ viết khóa bộ viết chỉ khi chúng cùng cập nhật một hàng
 - Hiệu suất giống với đọc cam kết
 - Không có đọc bản, đọc không lặp, hoặc đọc ảo tưởng
- Nhược điểm:
 - Hệ thống phải ghi và giữ những phiên bản cũ của các dữ liệu đã thay đổi
(Chỉ có ngày tháng thay đổi giữa việc bắt đầu và kết thúc của giao dịch chi đọc)
 - Không đảm bảo tính tuần tự cho các giao dịch đọc/ghi
- Ví dụ bổ xung: Oracle 9i
 - Không chi phí (no overhead): Tác động đến tiền ảnh (before-image) trong rollback segment
 - Thời gian hết hiệu lực của cấu hình các tiền ảnh (before-images configurable), “ảnh chụp quá cũ” bị thất bại nếu giá trị này quá nhỏ

Ảnh chụp tách bạch – Kết luận

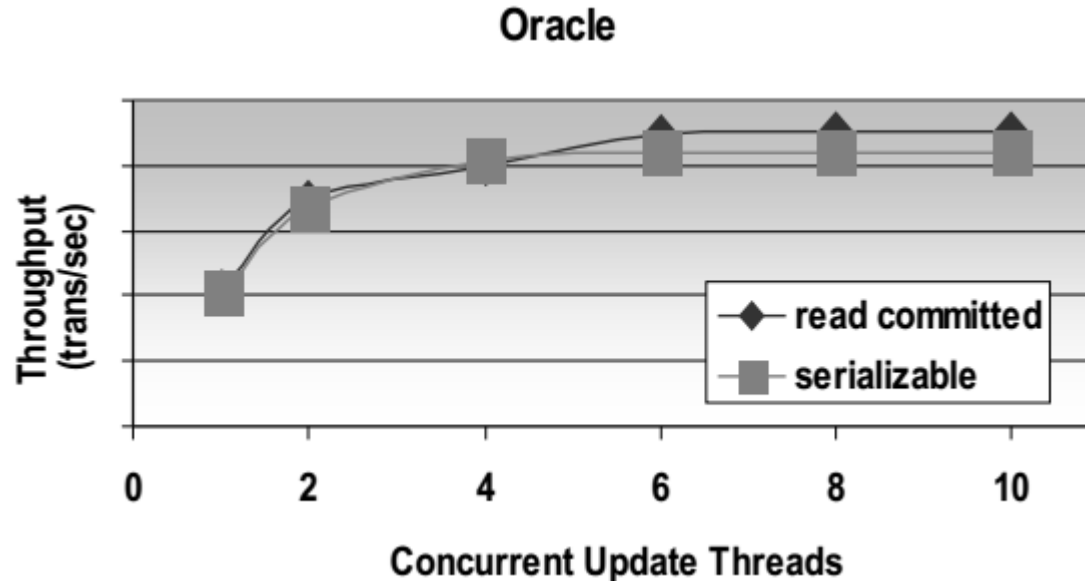
- Các ưu điểm của hiệu suất rất đáng kể do việc đọc không bị chặn và không chặn các giao dịch khác
- Không hoàn toàn tuần tự, mặc dù không có đọc bản, không lặp, hoặc ảo tưởng

Thực nghiệm: Đọc cam kết và tuần tự (Read Commit vs. Serializable)



- Truy vấn lấy tổng với các hoạt động chuyển khoản giữa các tài khoản ngân hàng.
- Ảnh chụp tách bạch Oracle (Read Commit vs. Serializable): Truy vấn lấy tổng chỉ đọc không bị nhiễu bởi các truy vấn chuyển khoản đồng thời
- Truy vấn lấy tổng (chỉ đọc) luôn cho kết quả chính xác
-

Thực nghiệm: Đọc cam kết và tuần tự (Read Commit vs. Serializable)



- Cả “read commit” (Đọc cam kết) và “serializable” (Tuần tự) sử dụng ảnh chụp tách bạch
- “serializable” hủy các giao dịch trong trường hợp việc ghi bị xung đột
- Các truy vấn lấy tổng luôn cho những câu trả lời đúng