

Quản trị cơ sở dữ liệu và Tối ưu hiệu năng

Tối ưu tương tranh
(concurrency tuning)

Nội dung

- Tối ưu chỉ số (Index tuning)
 - Những ví dụ về tối ưu chỉ số
- Tối ưu tương tranh (Concurrency Tuning)
 - Từ giới thiệu đến các giao dịch (Transactions)

Danh mục

- **Tối ưu chỉ số (Index tuning)**
 - Những ví dụ về tối ưu chỉ số
- Tối ưu tương tranh (Concurrency Tuning)
 - Từ giới thiệu đến các giao dịch (Transactions)

Các ví dụ về tối ưu chỉ số

- Các ví dụ sử dụng những bảng sau:
 - Employee(ssnum,name,dept,manager,salary)
 - Student(ssnum,name,course,grade,stipend,evaluation)

Thực hành 1 – Truy vấn cho Student bằng Name

- Bảng Student đã được tạo ra với chỉ số phi phân nhóm (non-clustering index) trên thuộc tính name.
- Truy vấn:

```
SELECT *  
FROM Student  
WHERE name='Bayer'
```
- Vấn đề: Truy vấn không sử dụng chỉ số (index) trên thuộc tính name
- Giải pháp: Thử cập nhật những số liệu thống kê danh mục:
 - Oracle, Postgres: ANALYZE
 - SQL Server: sp_createstats
 - DB2: RUNSTATS

Thực hành 2 - Truy vấn cho Salary I

- Chỉ số phi phân nhóm trên thuộc tính salary (mức lương).
- Số liệu thống kê danh mục được cập nhật.
- Truy vấn:
SELECT *
FROM Employee
WHERE salary/12 = 4000
- Vấn đề: Truy vấn quá chậm.
- Giải pháp: chỉ số không được sử dụng vì **biểu thức số học**.

Có 2 tùy chọn:

- Viết lại truy vấn:

```
SELECT *  
FROM Employee  
WHERE salary = 48000
```

- Sử dụng hàm số dựa trên chỉ số

Thực hành 3 - Truy vấn cho Salary II

- Chỉ số phi phân nhóm trên thuộc tính salary.
- Số liệu thống kê danh mục được cập nhật.
- Truy vấn:
SELECT *
FROM Employee
WHERE salary = 48000
- Vấn đề: Truy vấn vẫn không sử dụng chỉ số. Vậy lý do là gì?
- Giải pháp: Chỉ số là phi phân nhóm. Nhiều employee có salary = 48000, vì vậy chỉ số không thể giúp được ở trường hợp này, nó có thể giúp trong trường hợp khác, ít thường xuyên hơn.

Thực hành 4 - Chỉ số phân nhóm và những dư thừa

- Chỉ số phân nhóm trên Student.ssnun
- Kích thước trang: 2kB
- Kích thước bản ghi trong bảng Student: 1KB (Các giá trị của thuộc tính evaluation là một văn bản dài)
- Vấn đề: Dư thừa khi những evaluation mới được thêm vào.
- Giải pháp: Chỉ số phân nhóm không hỗ trợ được nhiều do kích thước bản ghi lớn. Một chỉ số phi phân nhóm sẽ tránh được những dư thừa.

Thực hành 5 – Chỉ số phi phân nhóm I

- Bảng employee:
 - 30 bản ghi employee trên một trang
 - Mỗi employee thuộc một trong 50 phòng ban (dept)
 - Mỗi phòng ban có kích thước giống nhau
- Truy vấn:

```
SELECT ssn  
FROM Employee  
WHERE dept = 'IT'
```
- Vấn đề: Chỉ số phi phân nhóm trên Employee.dept có giúp được không?
- Giải pháp: : Chỉ khi chỉ số bao trùm (covers) lên truy vấn
 - $30/50=60\%$ trang sẽ có một bản ghi với dept = 'IT'
 - Duyệt bảng (table scan) nhanh hơn việc truy cập đến 3/5 trang với thứ tự ngẫu nhiên

Thực hành 6 – Chỉ số phi phân nhóm II

- Bảng employee:
 - 30 bản ghi employee trên một trang
 - Mỗi employee thuộc một trong 5000 phòng ban (dept)
 - Mỗi phòng ban có kích thước giống nhau
- Truy vấn:

```
SELECT ssn  
FROM Employee  
WHERE dept = 'IT'
```
- Vấn đề: Chỉ số phi phân nhóm trên Employee.dept có giúp được không?
- Giải pháp: Chỉ khi chỉ số bao trùm lên truy vấn
 - Chỉ $30/5000=0.06\%$ trang sẽ có một bản ghi với dept = 'IT'
- Duyệt bảng sẽ chậm hơn

Thực hành 7 – Phân tích thống kê

- Các kiểm toán viên thực hiện một phân tích thống kê trên một bản sao của Employee.
- Truy vấn:
 - Đếm các nhân viên với một mức lương (salary) nhất định (thường xuyên)
 - Tìm các nhân viên với mức lương lớn nhất hoặc nhỏ nhất trong một phong ban cụ thể (thường xuyên)
 - Tìm một nhân viên bằng số an sinh xã hội (social security number) (hiếm khi)
- Vấn đề: Những chỉ số nào được tạo ra?
- Giải pháp:
 - Chỉ số phi phân nhóm trên thuộc tính salary (vì bao trùm lên truy vấn)
 - Chỉ số tổng hợp phân nhóm (clustering composite index) trên (dept, salary) sử dụng một B⁺-tree (tất cả nhân viên với mức lương lớn nhất ở trên các trang liên tiếp)
 - Chỉ số băm phi phân nhóm (non-clustering hash index) trên thuộc tính ssn.

Thực hành 8 – Các biểu thức đại số

- Tiềm thù lao (stipend) của học sinh là hàng tháng, tiền lương của nhân viên là hàng năm.
- Truy vấn: Những nhân viên và học sinh nào được trả lương nhiều như nhau.
- Có 2 tùy chọn để viết truy vấn:

<pre>SELECT * FROM Employee, Student WHERE salary = 12*stipend</pre>	<pre>SELECT * FROM Employee, Student WHERE salary/12 = stipend</pre>
--	--
- Chỉ số trên một bảng với một biểu thức đại số không được sử dụng.
- Vấn đề: Truy vấn nào tốt hơn?

Thực hành 8 – Giải pháp

- Nếu chỉ số chỉ trên **một bảng**, thì nó nên được sử dụng.
- Chỉ số trên **cả 2 bảng**, phân nhóm (clustering) trên bảng lớn hơn: sử dụng nó
- Chỉ số trên **cả 2 bảng**, phi phân nhóm (non - clustering) trên bảng lớn hơn:
 - Bảng nhỏ có (nhiều) ít bản ghi hơn bảng lớn có các trang: sử dụng chỉ số trên bảng lớn.
 - Ngược lại: Sử dụng chỉ số trên bảng nhỏ.

Thực hành 9 – Phòng thu mua (Purchasing Department)

- Phòng thu mua có bảng sau:
Onorder(supplier,part,quantity,price).
- Bảng này được sử dụng nhiều trong các giờ mở cửa, nhưng không đầu đêm.
- Truy vấn:
 - Q1: Thêm một bản ghi, với tất cả các trường đã được xác định (rất thường xuyên)
 - Q2: Xóa một bản ghi, với thuộc tính supplier và part được xác định (rất thường xuyên)
 - Q3: Tìm tổng số lượng các part đã cho trên order (thường xuyên)
 - Q4: Tìm toàng bộ giá trị trên order với một supplier đã cho (hiếm khi)
- Vấn đề: Những chỉ số nào nên được sử dụng?

Thực hành 9 – Giải pháp

- Truy vấn:
 - Q1: Thêm một bản ghi, với tất cả các trường đã được xác định (rất thường xuyên)
 - Q2: Xóa một bản ghi, với thuộc tính supplier và part được xác định (rất thường xuyên)
 - Q3: Tìm tổng số lượng các part đã cho trên order (thường xuyên)
 - Q4: Tìm toàn bộ giá trị trên order với một supplier đã cho (hiếm khi)
- Giải pháp: Chỉ số B⁺-tree tổng hợp phân nhóm (Clustering composite B⁺-tree index) trên (part,supplier).
 - Loại bỏ những dư thừa khi sử dụng bảng trên qua đêm.
 - Order quan trọng để hỗ trợ truy vấn Q3
 - Chỉ số băm (hash index) sẽ không làm việc với truy vấn Q3 (Tiềm tố trên Q3 phù hợp với truy vấn)
- Thảo luận: Chỉ số phi phân nhóm trên thuộc tính supplier có để trả lời cho truy vấn Q4 không?
 - Chỉ số phải được duy trì và sẽ gây ảnh hưởng thường xuyên đến hiệu năng của các truy vấn Q1 và Q2.
 - Chỉ số không giúp nhiều nếu chỉ có một vài supplier khác biệt.

Thực hành 10 - Truy vấn điểm quá chậm (Point Query Too Slow)

- Employee có một chỉ số B⁺-tree phân nhóm trên ssnum.
- Truy vấn:
 - Lấy employee bởi số an sinh xã hội (ssnum).
 - Cập nhật employee với số an sinh xã hội
- Vấn đề: Thông lượng (Throughput) vẫn chưa đủ.
- Giải pháp: Sử dụng chỉ số băm thay vì B⁺-tree (Nhanh hơn cho các truy vấn điểm).

Thực hành 11 – Cơ sở dữ liệu về lịch sử nhập cư

- Cơ sở dữ liệu số hóa (Digitalized database) của những người nhập cư Tại Mỹ (US immigrants) giữa 1800 và 1900:
 - 17M các bản ghi.
 - Mỗi bản ghi có khoảng 200 trườngVí dụ: last name, first name, city of origin, ship taken,...
- Truy vấn: Lấy những người nhập cư:
 - Bởi last name và ít nhất một thuộc tính khác.
 - Thuộc tính thứ 2 thường là tên đầu first name (thường xuyên nhất) hoặc năm.
- Vấn đề: Đáp ứng một cách hiệu quả 2M các con cháu của các người nhập cư...

Thực hành 11 – Giải pháp

- Chỉ số B⁺-tree phân nhóm trên (lastname,firstname):
 - Không có dư thừa vì CSDL không có cập nhật
 - Sử dụng yếu tố đáp ứng cao để tăng không gian sử dụng
 - Nén khóa (key compression) nên được sử dụng (khóa dài, không cập nhật)
 - Chỉ số cũng hữu ích cho những truy vấn tiền tố (prefix queries) trên lastname
- Chỉ số phi phân nhóm kết hợp (Composite non-clustering index) trên (lastname,year):
 - Không có chi phí duy trì (không cập nhật)
 - Các thuộc tính có thể có đủ tính chọn lọc (selective enough)
- Những chỉ số phi phân nhóm có luôn trên tất cả các tổ hợp thuộc tính không?
 - Không cần duy trì, vì vậy giới hạn chỉ là chi phí không gian (space overhead)
 - Hữu ích chỉ khi có đủ tính chọn lọc (selective enough)

Thực hành 12 – Hệ thống đặt trước chuyến bay

- Một công ty hàng không quản lý 1000 chuyến bay và sử dụng bảng sau:
 - Flight(flightID, seatID, passanger-name)
 - Totals(flightID, number-of-passangers)
- Truy vấn:Mỗi đơn đặt trước:
 - Thêm một bản ghi vào bảng Flight
 - Tăng Totals.number-of-passangers
- Các truy vấn là các giao dịch riêng rẽ
- Vấn đề: Ngăn ngừa sự xung đột trên bảng Totals.
- Giải pháp:
 - Totals là một bảng nhỏ (1000 bản ghi nhỏ) và phù hợp trên một vài trang.
 - Không có chỉ số, sự cập nhật sẽ duyệt bảng, các bản ghi đã duyệt sẽ bị khóa
 - Chỉ số phân nhóm trên thuộc tính flightID tránh được duyệt bảng, vì vậy sẽ chặn được xung đột (row locking assumed).

Danh mục

- Tối ưu chỉ số (Index tuning)
 - Những ví dụ về tối ưu chỉ số
- **Tối ưu tương tranh (Concurrency Tuning)**
 - **Từ giới thiệu đến các giao dịch (Transactions)**

Giao dịch là gì?

- Một giao dịch (Transaction) là một đơn vị của sự thực hiện chương trình để truy cập và có thể cập nhật các mục dữ liệu (data items) khác nhau.
- Ví dụ: Chuyển 50\$ từ tài khoản A đến tài khoản B
 1. R (A)
 2. $A \leftarrow A - 50$
 3. W (A)
 4. R (B)
 5. $B \leftarrow B + 50$
 6. W (B)
- 2 vấn đề chính:
 - Sự thực hiện đồng thời của nhiều giao dịch
 - Các loại sự cố khác nhau (ví dụ: sự cố phần cứng, hệ thống bị treo)

Các thuộc tính ACID

- Hệ thống CSDL phải đảm bảo ACID cho các giao dịch:
 - Atomicity (Tính nguyên tử): Hoặc tất cả các hoạt động của giao dịch được thực hiện hoặc không.
 - Consistency (Tính nhất quán): Sự thực hiện của giao dịch trong sự tách biệt duy trì tính nhất quán của CSDL.
 - Isolation (Tính tách biệt): Mặc dù nhiều giao dịch có thể thực hiện đồng thời, nhưng mỗi giao dịch phải không nhận thấy những giao dịch đồng thời khác.
 - Durability (Tính bền): Sau khi một giao dịch thành công, Những thay đổi CSDL vẫn tồn tại ngay cả trong trường hợp sự cố hệ thống.

Tính nguyên tử (Atomicity)

- Ví dụ: Chuyển \$50 từ tài khoản A đến tài khoản B
 1. R (A)
 2. $A \leftarrow A - 50$
 3. W (A)
 4. R (B)
 5. $B \leftarrow B + 50$
 6. W (B)
- Điều gì sẽ xảy ra nếu thất bại (phần cứng hoặc phần mềm) sau bước 3?
 - Tiền bị mất
 - CSDL không nhất quán
- Tính nguyên tử:
 - Hoặc thực hiện tất cả hoạt động hoặc không
 - Những cập nhật một cách từng phần của các giao dịch đã được thực hiện không phản ánh trong CSDL.

Tính nhất quán (Consistency)

- Ví dụ: Chuyển \$50 từ tài khoản A đến tài khoản B
 1. R (A)
 2. $A \leftarrow A - 50$
 3. W (A)
 4. R (B)
 5. $B \leftarrow B + 50$
 6. W (B)
- Tính nhất quán trong ví dụ: Tổng A+B phải không thay đổi
- Tính nhất quán nói chung:
 - Các ràng buộc toàn vẹn rõ ràng (explicit integrity constraints) (ví dụ: khóa ngoài)
 - Các ràng buộc toàn vẹn tiềm ẩn (implicit integrity constraints) (ví dụ: tổng của tất cả các số dư tài khoản của một chi nhánh ngân hàng phải bằng với số dư của trị nhánh)
- Giao dịch:
 - Phải nhận thấy cơ sở dữ liệu nhất quán (consistent database)
 - Trong khi trạng thái mâu thuẫn giao dịch (transaction inconsistent state) được cho phép
 - Sau khi hoàn thành, CSDL phải được thống nhất lại

Tính tách biệt (Isolation)

- Ví dụ: Chuyển \$50 từ tài khoản A đến tài khoản B
 1. R (A)
 2. $A \leftarrow A - 50$
 3. W (A)
 4. R (B)
 5. $B \leftarrow B + 50$
 6. W (B)
- Hình dung giao dịch thứ hai T2:
 - T2: R (A), R (B), print (A + B)
 - T2 được thực hiện giữa bước 2 và 4
 - T2 nhận thấy một CSDL không phù hợp (inconsistent database) và trả lại kết quả sai

Tính tách biệt (Isolation)

- Tính tách biệt thông thường (Trivial isolation): Thực hiện giao dịch tuần tự
- Tính tách biệt cho các giao dịch đồng thời: Với mỗi cặp giao dịch T_i và T_j , xuất hiện T_i khi hoặc T_j kết thúc việc thực hiện trước khi T_i bắt đầu, hoặc T_j bắt đầu thực hiện sau khi T_i kết thúc.
- Biểu thời gian (Schedule):
 - Xác định thứ tự thời gian (chronological order) của một chuỗi các chỉ dẫn (instructions) từ các giao dịch khác nhau.
 - Các biểu thời gian tương đương (equivalent schedules) dẫn tới những CSDL đồng nhất (identical databases) nếu chúng bắt đầu với những CSDL đồng nhất.
- Biểu thời gian tuần tự (Serializable schedule):
 - Tương đương với một số biểu thời gian nối tiếp (serial schedule)
 - Biểu thời gian tuần tự của T_1 và T_2 tương đương với T_1, T_2 hoặc T_2, T_1

Tính bền (Durability)

- Khi một giao dịch kết thúc, nó sẽ cam kết (commit)
- Ví dụ: Giao dịch cam kết quá sớm
 - Giao dịch viết A, sau đó cam kết
 - A được ghi vào bộ đĩa đệm
 - Sau đó hệ thống bị treo
 - Giá trị của A bị mất
- Tính bền: Sau khi một giao dịch đã được cam kết, các thay đổi của CSDL vẫn tồn tại ngay cả trong trường hợp lỗi hệ thống.
- Cam kết: Chỉ sau khi tất cả thay đổi đã cố định (permanent):
 - Hoặc ghi vào log file hoặc trực tiếp đến CSDL
 - CSDL phải được thu hồi trong trường hợp gặp sự cố

Khóa (Locks)

- Một khóa là một cơ chế điều khiển tương tranh (control concurrency) trên một mục dữ liệu (data item).
- 2 kiểu khóa trên một mục dữ liệu A là:
 - Đặc biệt (exclusive) - $xL(A)$: Mục dữ liệu A có thể được đọc và được ghi.
 - Chia sẻ (shared) - $sL(A)$: Mục dữ liệu A chỉ được đọc.
- Yêu cầu khóa (Lock request) được tạo ra để quản lý điều khiển tương tranh
- Giao dịch bị chặn (blocked) cho đến khi khóa được cấp.
- Mở khóa A (Unlock A) – $uL(A)$: Giải phóng khóa trên một mục dữ liệu A

Tính tương thích khóa (Lock Compatibility)

Ma trận tương thích khóa:

$T_1 \downarrow T_2 \rightarrow$	shared	exclusive
shared	true	false
exclusive	false	false

- T_1 giữ khóa chia sẻ (shared lock) trên A:
 - Khóa chia sẻ được cấp cho T_2
 - Khóa đặc biệt không được cấp cho T_2
- T_2 giữ khóa đặc biệt (exclusive lock) trên A:
 - Khóa chia sẻ không được cấp cho T_2
 - Khóa đặc biệt không được cấp cho T_2
- Các khóa chia sẻ có thể được chia sẻ bởi bất kì số lượng các giao dịch

Giao thức khóa (Locking Protocol)

- Ví dụ giao dịch T_2 với khóa (locking):
 1. sL(A), R (A), uL(A)
 2. sL(B), R (B), uL(B)
 3. print (A + B)
- T_2 sử dụng khóa, nhưng không có tính tuần tự (not serializable)
 - A hoặc/và B có thể được cập nhật giữa bước 1 và 2
 - In ra tổng có thể sai.
- Giao thức khóa:
 - Là một tập các luật được xây dựng dựa theo tất cả các giao dịch khi yêu cầu/giải phóng các khóa.
 - Giao thức khóa giới hạn tập các biểu thời gian có thể.

Các bẫy của các giao thức khóa – Bế tắc (Pitfalls of Locking Protocols – Deadlock)

- Ví dụ:
 - T_1 : $R(A)$, $A \leftarrow A + 10$, $R(B)$, $B \leftarrow B - 10$, $W(A)$, $W(B)$
 - T_2 : $R(B)$, $B \leftarrow B + 50$, $R(A)$, $A \leftarrow A - 50$, $W(A)$, $W(B)$
 - Kịch bản tương tranh có thể với các khóa:
 $T1.xL(A)$, $T1.R(A)$, $T2.xL(B)$, $T2.R(B)$, $T2.xL(A)$, $T1.xL(B)$, ...
 - T_1 và T_2 chặn lẫn nhau – có thể không có tiến triển
- Bế tắc (Deadlock): Tình trạng các giao dịch chặn lẫn nhau
- Xử lý bế tắc:
 - Một trong các giao dịch phải được phục hồi (roll back)
 - Giao dịch đã phục hồi giải phóng khóa

Các bẫy của các giao thức khóa – Chết tắc

- Chết tắc (Starvation): giao dịch tiếp tục đợi khóa.
- Các ví dụ:
 - Giao dịch giống nhau lặp đi lặp lại việc phục hồi do bế tắc
 - Một giao dịch tiếp tục đợi một khóa đặc biệt (exclusive lock) trên một mục CSDL trong khi một chuỗi các giao dịch khác được cấp các khóa chia sẻ (shared locks).
- Bộ quản lý tương tranh thiết kế tốt (Well-designed concurrency manager) để tránh chết tắc.

Hai giai đoạn khóa (Two-Phase Locking)

- Giao thức phải đảm bảo tính tuần tự (serializability):
- Giai đoạn 1: Giai đoạn phát triển
 - Giao dịch có thể thu được các khóa
 - Giao dịch có thể không giải phóng các khóa
- Giai đoạn 2: Giai đoạn thu hẹp
 - Giao dịch có thể giải phóng các khóa
 - Giao dịch không thể thu được các khóa

Hai giai đoạn khóa – Ví dụ

- Ví dụ: Hai giao dịch chuyển tiền đồng thời:
T1: R (A), $A \leftarrow A + 10$, R (B), $B \leftarrow B - 10$, W (A), W (B)
T2: R (A), $A \leftarrow A - 50$, R (B), $B \leftarrow B + 50$, W (A), W (B)
- Biểu thời gian cho 2 giai đoạn khóa có thể:
 1. T1 : xL(A), xL(B), R (A), R (B), W ($A \leftarrow A + 10$), uL(A)
 2. T2 : xL(A), R (A), xL(B) (wait)
 3. T1 : W ($B \leftarrow B - 10$), uL(B)
 4. T2 : R (B), W ($A \leftarrow A - 50$), W ($B \leftarrow B + 50$), uL(A), uL(B)
- Biểu thời gian nối tiếp tương đương (Equivalent serial schedule): T_1, T_2