CS5226 Lecture 10
Review

Lecture topics

2 & 3) Query tuning

- Eliminating redundant DISTINCT
- Unnesting queries with apply operator
- Reordering group by with select/join

4) Index tuning - Microsoft SQL Server's approach
5) Materialized view tuning - Microsoft SQL Server's approach
6 & 7) Transaction tuning

- Transaction chopping
- Snapshot isolation tuning

8) Memory tuning - Oracle 9i's PGA tuning

9) Statistics tuning - DB2's use of statistical views

Final Exam

- Date: April 27 (Saturday)
- Time: 1pm
- Duration: 2 hrs
- Format:
  - Open book exam
  - Multiple-choice and short questions (calculator required)
- Scope: all topics covered except lecture 4
  - Lectures 2 & 3: Query tuning
  - Lecture 5: Materialized view tuning
  - Lecture 6: Transaction Chopping
  - Lecture 7: Snapshot Isolation Tuning
  - Lecture 8: Memory Tuning
  - Lecture 9: Statistics Tuning

Q1: Query Tuning

Consider the three relations R, S, and T (where the key attribute of each relation is underlined) and queries Q_1 and Q_2.

- R(A,B,C)
- S(D,E,F)
- T(G,H,I)

Q_1:

```
select   distinct C, D
from     R, S, T
where    E = I
and      C = G
and      B = 10
```

Q_2:

```
select   distinct A, G
from     R, S, T
where    B = H
and      D = 50
and      I = 100
```

Q1: Query Tuning (cont.)

- State whether the following statement is *true* or *false*: Based on the algorithm discussed in class, we can't conclude that the distinct clause in Q_1 is redundant.
- State whether the following statement is *true* or *false*: Based on the algorithm discussed in class, we can't conclude that the distinct clause in Q_2 is redundant.

Q_1:

```
select   distinct C, D
from     R, S, T
where    E = I
and      C = G
and      B = 10
```

Q_2:

```
select   distinct A, G
from     R, S, T
where    B = H
and      D = 50
and      I = 100
```

Q4: Snapshot Isolation Tuning

Let A be an application that consists of three transactional programs $\{P_1, P_2, P_3\}$ such that $SDG(A)$ does not contain any dangerous structure.
Suppose we create a new application A' from A by adding an additional program P_4; i.e., A' consists of four transactional programs $\{P_1, P_2, P_3, P_4\}$.

- If all the programs in A contain no update statements (i.e, they are all read-only programs), state whether the following statement is *true* or *false*:
  $SDG(A')$ does not contain any dangerous structure.

Q4: Snapshot Isolation Tuning (cont.)

- If P_4 does not contain any update statement (i.e., it is a read-only program), state whether the following statement is *true* or *false*:
  $SDG(A')$ does not contain any dangerous structure.

Q5: Snapshot Isolation Tuning

Consider a database consisting of three relations $R_1(\underline{K_1}, A_1)$, $R_2(\underline{K_2}, A_2)$, and $R_3(\underline{K_3}, A_3)$, where the key of each R_i is K_i. Consider an application that consists of the following four transactional programs, W, X, Y, and Z.

Program W(V)

```
SELECT A1 into :A
FROM R1
WHERE K1 = :V;

UPDATE R2
SET A2 = A2 + 1
WHERE K2 = :A;

COMMIT;
```

Program X(V)

```
UPDATE R2
SET A2 = A2 - 1
WHERE K2 = :V;

COMMIT;
```

Program Y(V)

```
SELECT A2 into :A
FROM R2
WHERE K2 = :V;

UPDATE R3
SET A3 = A3 - 2
WHERE K3 = :A;

COMMIT;
```

Program Z(V)

```
SELECT A2 into :A
FROM R2
WHERE K2 = :V;

SELECT A3 into :B
FROM R3
WHERE K3 = :V;

UPDATE R1
SET A1 = A1 + :B
WHERE K1 = :A;

COMMIT;
```

Q5: Snapshot Isolation Tuning (cont.)

- Show the static dependency graph for this application.
- State the minimum number of programs that need to be modified so that the application can become serializable under Snapshot Isolation. Support your answer by showing the necessary modifications for each program to be modified. Note that you only need to show the program statements (i.e., SELECT/UPDATE statements) that are modified; do not repeat any unmodified program statements.

Program W(V)

```
SELECT A1 into :A
FROM R1
WHERE K1 = :V;

UPDATE R2
SET A2 = A2 + 1
WHERE K2 = :A;

COMMIT;
```

Program X(V)

```
UPDATE R2
SET A2 = A2 - 1
WHERE K2 = :V;

COMMIT;
```

Program Y(V)

```
SELECT A2 into :A
FROM R2
WHERE K2 = :V;

UPDATE R3
SET A3 = A3 - 2
WHERE K3 = :A;

COMMIT;
```

Program Z(V)

```
SELECT A2 into :A
FROM R2
WHERE K2 = :V;

SELECT A3 into :B
FROM R3
WHERE K3 = :V;

UPDATE R1
SET A1 = A1 + :B
WHERE K1 = :A;

COMMIT;
```

Q6: Xact Chopping

Consider the following set of transactions $S = \{T_1, T_2, T_3, T_4\}$:

$$
\begin{aligned}
T_1&: \quad R_1(a), R_1(d), W_1(a), R_1(b), W_1(c) \\
T_2&: \quad R_2(d), W_2(d), R_2(e), W_2(e) \\
T_3&: \quad W_3(f), R_3(g), W_3(g) \\
T_4&: \quad W_4(a), R_4(e), R_4(g), W_4(e)
\end{aligned}
$$

Write down each of the following:

- FineChop(T_1)
- FineChop(T_2)
- FineChop(T_3)
- FineChop(T_4)

Q7: Xact Chopping

Let chop(T_1), chop(T_2), and chop(T_3) denote a rollback-safe chopping of transactions T_1, T_2, and T_3, respectively.

Let G_{12} denote the chopping graph consisting of $\{chop(T_1), chop(T_2)\}$, G_{13} denote the chopping graph consisting of $\{chop(T_1), chop(T_3)\}$, G_{23} denote the chopping graph consisting of $\{chop(T_2), chop(T_3)\}$, and G_{123} denote the chopping graph consisting of $\{chop(T_1), chop(T_2), chop(T_3)\}$.

Q7: Xact Chopping (cont.)

(i) Which of the following statements is the most appropriate about G_{12}, G_{13}, G_{23}, and G_{123}?

- (a) If each of G_{12}, G_{13}, and G_{23} does not contain any SC-cycle, then G_{123} does not contain any SC-cycle.
- (b) If G_{123} does not contain any SC-cycle, then each of G_{12}, G_{13}, and G_{23} does not contain any SC-cycle.
- (c) (a) or (b)
- (d) (a) and (b)
- (e) None of the above.

(ii) Justify your answer in (i)