Gestione dei dati

Parte 7
Esercitazione sulla valutazione delle query

Maurizio Lenzerini, Riccardo Rosati

Facoltà di Ingegneria
Sapienza Università di Roma
Anno Accademico 2012/2013

http://www.dis.uniroma1.it/~rosati/gd/

SAPIENZA
UNIVERSITÀ DI ROMA

Esercizio 1 (valutazione query)

Si consideri una base di dati contenente la relazione Studente con attributi Matricola, Cognome, Nome, DataNascita, CodiceComune, e la relazione Comune con attributi CodiceComune, NomeComune, NumeroAbitanti, Provincia, CAP. Si assuma che la relazione Studente contenga 100000 record e che ogni record di tale relazione abbia dimensione P/40, dove P è la dimensione di una pagina di memoria. Inoltre si assuma che la relazione Comune contenga 1000 record e che ogni record di tale relazione abbia dimensione P/50.

Si consideri ora la query:

```
select S.Cognome, S.Nome, C.NumeroAbitanti
from Studente S, Comune C
where S.CodiceComune = C.CodiceComune
```

Calcolare il costo della valutazione della query nelle seguenti ipotesi:

1. il sistema esegue l'algoritmo naive nested loop;
2. il sistema esegue l'algoritmo nested loop;
3. il sistema esegue l'algoritmo block nested loop, assumendo 12 pagine disponibili del buffer.

Soluzione esercizio 1

1) Costo dell'algoritmo naive nested loop: $M + (p_R \times M \times N)$

con M = numero pagine outer relation

N= numero pagine inner relation

p_R = numero record outer relation

scegliamo Comune come outer relation (perché è la relazione più piccola), pertanto

M= 1000 / 50 = 20

N= 100000 / 40 = 2500

p_R = 1000 / 20 = 50

il costo è quindi 20 + (50 x 20 x 2500) = 20 + 2,5 x 10^6

Soluzione esercizio 1

2) Costo dell'algoritmo nested loop: $M + (M \times N)$

con M = numero pagine outer relation

N= numero pagine inner relation

scegliamo ancora Comune come outer relation, pertanto

M= 1000 / 50 = 20

N= 100000 / 40 = 2500

il costo è quindi 20 + (20 x 2500) = 50020

Soluzione esercizio 1

3) Costo dell'algoritmo block nested loop: $M + (N \times M/(G-2))$

con M = numero pagine outer relation

N= numero pagine inner relation

G = numero pagine buffer disponibili

scegliamo ancora Comune come outer relation, pertanto

M= 1000 / 50 = 20

N= 100000 / 40 = 2500

G = 12

il costo è quindi 20 + (2500 x 20 / 10) = 5020

Esercizio 2

Si consideri una base di dati contenente la relazione Studente con attributi Matricola, Cognome, Nome, DataNascita, CodiceComune, e la relazione Comune con attributi CodiceComune, NomeComune, NumeroAbitanti, Provincia, CAP. Si assuma che la relazione Studente contenga 100000 record e che ogni record di tale relazione abbia dimensione P/40, dove P è la dimensione di una pagina di memoria. Inoltre si assuma che la relazione Comune contenga 1000 record e che ogni record di tale relazione abbia dimensione P/50. Si assuma infine che il sistema esegua i join utilizzando l'algoritmo Block Nested Loop.

Si consideri ora la query:

```
select S.Cognome, S.Nome, C.NumeroAbitanti
from Studente S, Comune C
where S.CodiceComune = C.CodiceComune
```

1. Dire quali organizzazioni di file sono le più indicate per le relazioni Studente e Comune, motivando la risposta;
2. assumendo che il buffer possa dedicare 7 pagine all'esecuzione di questa query, dire qual è il costo dell'esecuzione di tale query.

Soluzione esercizio 2

Rsiposta alla domanda 1:

- la query è un equi join
- è sempre preferibile scegliere la relazione più piccola, cioè Comune, come outer relation del join. Per tale relazione non è necessaria nessuna organizzazione di file particolare, in quanto il Block Nested Loop effettua una scansione della outer relation (e per eseguire l'operazione di scansione è comunque necessario leggere tutto il file), per cui è sufficiente anche un heap file
- anche per la inner relation l'algoritmo Block Nested Loop effettua una semplice scansione, per cui anche per la relazione Studente non è in realtà necessaria alcuna organizzazione di file particolare (è sufficiente un heap file)

Soluzione esercizio 2 (continua)

Risposta alla domanda 2:

- M = numero di pagine del file contenente la relazione Comune = 1000/50 = 20
- N = numero di pagine del file contenente la relazione Studente = 100000/40 = 2500
- G = pagine disponibili del buffer = 7
- costo del Block Nested Loop = M + (N × M/(G-2)) = 20 + (2500 × 4) = 10020
- (costo del Nested Loop = M + (M × N) = 20 + 50000 = 50020)

Esercizio 3

Sia data la seguente query:

SELECT R.A, S.B

FROM R, S

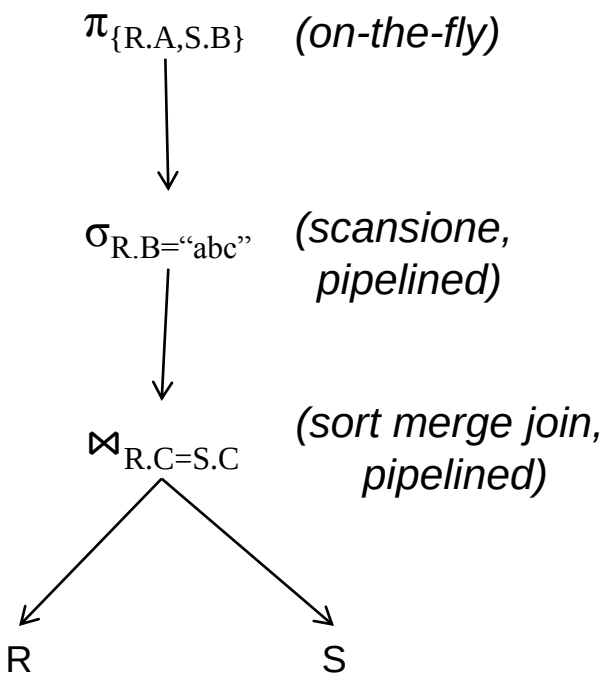
WHERE R.B=“abc” AND R.C = S.C

e si assuma che sulla relazione R sia dichiarato un indice hash con chiave di ricerca B, che sulla relazione S sia dichiarato un indice hash con chiave di ricerca C, che il DBMS possa eseguire i join sia mediante l’algoritmo sort merge join che mediante l’algoritmo block nested loop.

1) Scrivere due query plan per tale query.
2) Si assuma che R sia contenuto in 200 pagine, S sia contenuta in 100 pagine, che i record di R con B=“abc” siano contenuti in 1 pagina e che il DBMS abbia a disposizione 10 pagine per eseguire i join. Valutare il costo dell’esecuzione dei due query plan (espresso come numero di trasferimenti di pagine da memoria di massa).

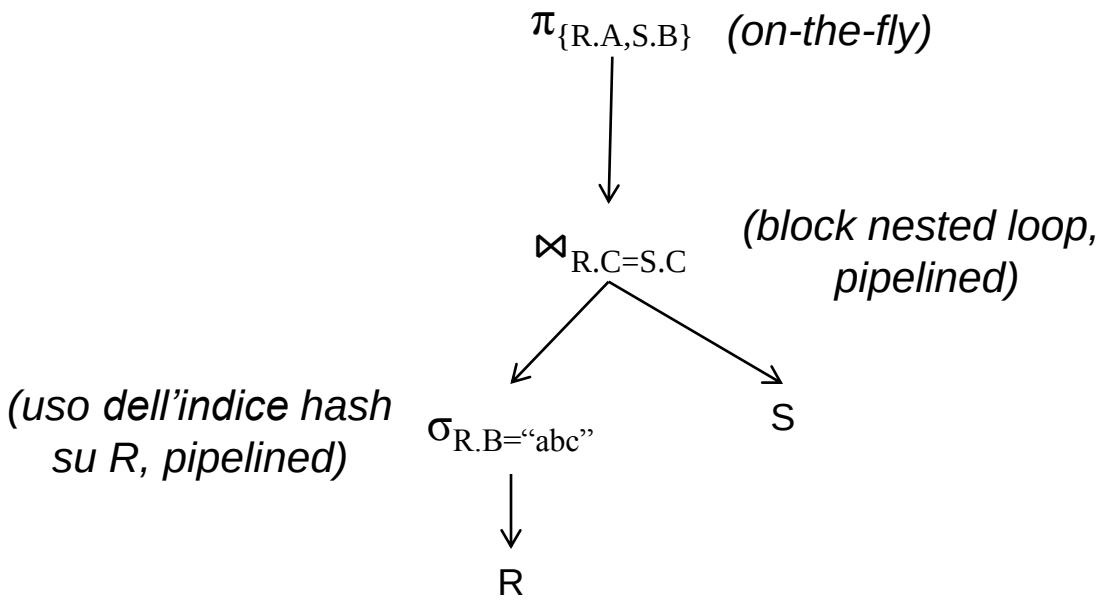
Soluzione esercizio 3

Primo query plan QP1:



Soluzione esercizio 3

Secondo query plan QP2:



Soluzione esercizio 3

Costo dell'esecuzione di QP1:

1) Sort merge join di R e S = 100 $\log_{10}$ 100 + 200 $\log_{10}$ 200 + 100 + 200 =200 + 600 + 100 + 200 = 1100 pagine trasferite da memoria di massa
2) Selezione sul risultato di 1) = nessun trasferimento di pagina da memoria di massa (pipelined)
3) Prioezione sul risultato di 2) = nessun trasferimento di pagina da memoria di massa (pipelined)

Numero totale di trasferimenti di pagine da memoria di massa = 1100

Soluzione esercizio 3

Costo dell'esecuzione di QP2:

1) Selezione su R tramite indice hash = 1 pagina trasferita da memoria di massa
2) Block nested loop tra il risultato di 1) e S = 100 = 100 pagine trasferite da memoria di massa
3) Proiezione sul risultato di 2) = nessun trasferimento di pagina da memoria di massa (pipelined)

Numero totale di trasferimenti di pagine da memoria di massa = 1+100 = 101

Pertanto QP2 risulta più efficiente di QP1.

Esercizio proposto

Si considerino le relazioni e la query dell'esercizio 1 e si assuma che il sistema esegua l'algoritmo Sort Merge Join. Dire quali organizzazioni di file sono preferibili per le relazioni Studente e Comune, e qual è il costo dell'esecuzione della query, assumendo che CodiceComune sia chiave della relazione Comune