Tecnologie delle Basi di Dati M

Appello del 16/9/2011

Esercizio 1 (2 punti)

Data la relazione con schema:

`ContoCorrente(codice, cognome, nome, indirizzo, saldo)`

si effettui una stima del numero di livelli (e di nodi) di un B^+-tree clustered costruito sull'attributo composto (`cognome, nome`), che è da considerarsi attributo chiave. In particolare, si distinguano i casi di indice *denso* e *sparso*. Si supponga di avere pagine di dimensione 4 KB, di cui 96 B riservati per il page header, e si considerino i seguenti valori:

- Numero di tuple = 5.4M
- Numero di pagine = 15K
- Dimensione `cognome` = dimensione `nome` = 23 byte
- Dimensione RID = 4 byte
- Dimensione PID = 2 byte
- Percentuale di riempimento foglie/nodi = 90%

Esercizio 2 (5 punti)

Data la relazione con schema:

`Personale(matricola, nome, data, luogo, stipendio, responsabile)`

si ottimizzi l'esecuzione della seguente interrogazione SQL:

```
SELECT P.matricola, R.matricola
FROM Personale P, Personale R
WHERE P.responsabile=R.matricola
  AND R.stipendio < 60000
  AND (P.luogo = 'Milano' OR P.luogo = 'Bologna')
```

tenendo conto che dai cataloghi della base di dati risulta:

- Numero di tuple `Personale` = 50K
- Numero di pagine `Personale` = 2K
- Numero di responsabili = 100
- Indice unclustered (TID ordinate) su `luogo`: numero foglie = 500, numero chiavi = 50
- Indice clustered su `stipendio`: numero foglie = 200, valore minimo = 20000, valore massimo = 220000
- Indice unclustered su `matricola`: numero foglie = 3000

Si disegni infine l'albero corrispondente al piano di accesso di costo minimo e stimi il numero di risultati dell'interrogazione.

Suggerimento: per la formula di Cardenas si utilizzino i seguenti valori, validi per P = 2000:

R	Φ(R, P)
0	0
50	49.39237
100	97.56494
150	144.5478
200	190.3704
250	235.0614
300	278.6486
350	321.1594
400	362.6204
450	403.0574

R	Φ(R, P)
500	442.4958
550	480.9602
600	518.4747
650	555.0627
700	590.7472
750	625.5503
800	659.494
850	692.5994
900	724.8872
950	756.3776

R	Φ(R, P)
1000	787.0904
1050	817.0446
1100	846.2591
1150	874.7521
1200	902.5414
1250	929.6445
1300	956.0782
1350	981.859
1400	1007.003
1450	1031.526

Esercizio 3 (5 punti)

Si confrontino le diverse tecniche per la gestione di overflow in area primaria per le strutture hash (metodi di concatenamento/indirizzamento aperto), evidenziandone pregi e difetti.

Esercizio 4 (3 punti)

Gli algoritmi presentati per la valutazione efficiente di join sono adatti ad essere utilizzati nel caso in cui la condizione di join sia una condizione di uguaglianza. Si discuta sulla possibile estensione degli algoritmi *sort-merge join* e *hash join* per la valutazione efficiente di outer join (destro, sinistro e full).

Soluzione Esercizio 1

Dimensionamento indice denso:
Dimensione di ogni record (foglia) = 4 + 23 + 23 = **50B**
Dimensione "reale" foglia/nodo = (4096 – 96) x 0.90 = **3600B**
Numero di record per foglia = 3600/50 = **72**
Numero di foglie = 5.4M/72 = **75000**
Dimensione di ogni record (nodo interno) = 2 + 23 + 23 = **48B**
Numero di record per nodo interno = 3600/48 = **75**
Numero nodi livello 1 = 75000/75 = **1000**
Numero nodi livello 2 = 1000/75 = **14**
Numero nodi livello 3 = 14/75 = **1**
Il B^+-tree corrispondente si compone quindi di 4 livelli per un totale di 1015 nodi interni e 75000 foglie.
Dimensionamento indice sparso:
Dimensione di ogni record (foglia/nodo) = 2 + 23 + 23 = **48B**
Numero di record per foglia/nodo = 3600/48 = **75**
Numero di foglie = 15K/75 = **200**
Numero nodi livello 1 = 200/75 = **3**
Numero nodi livello 2 = 3/75 = **1**
Il B^+-tree corrispondente si compone quindi di 3 livelli per un totale di 4 nodi interni e 200 foglie.

Soluzione Esercizio 2

Selettività dei predicati:
Predicato su `stipendio` = (60000 – 20000)/(220000 – 20000) = **0.2**
Predicato su `luogo` = 1/50 = **0.02** per ogni valore di `luogo`
Predicato di join = 1/50K (chiave esterna)

Accesso a P:
Costo scan sequenziale = **2000**
Costo indice su `luogo`: 2 x (NL x 0.02 + Φ(NT x 0.02, NP)) = 2 x (500 x 0.02 + Φ(50K x 0.02, 2K)) = 2 x (10 + Φ(1K, 2K)) = 2 x (10 + 788) = **1596**
Numero tuple residue = 2 x NT x 0.02 = **2000**

Accesso a R:
Costo scan sequenziale = **2000**
Costo indice su `stipendio`: NL x 0.2 + NP x 0.2 = 200 x 0.2 + 2K x 0.2 = 40 + 400 = **440**
Costo indice su `matricola`: 1 + 1 = **2**
Numero tuple residue = NT x 0.2 = **10000**

Costi di join:
`P` esterna: costo = costo indice `luogo` + 2000 x costo indice `matricola` = 1596 + 2000 x 2 = **5596**
`R` esterna: costo = costo indice `stipendio` + 10000 x costo indice `luogo` = 440 + 10000 x 1596 = **15960440**
Il numero di risultati dell'interrogazione è 50K x 0.04 x 0.2 = **400**