Tecnologie delle Basi di Dati M

Appello del 18/1/2012

Esercizio 1 (2 punti)

Data la relazione con schema:

`Personale(codice, cognome, nome, annodinascita, stipendio)`

si effettui una stima del numero di livelli (e di nodi) di un R-tree costruito sulla coppia di attributi (`annodinascita, stipendio`). 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 = 20K
- Numero di pagine = 100
- Dimensione `annodinascita` = dimensione `stipendio` = 8 byte
- Numero di chiavi (`annodinascita`) = 25
- Numero di chiavi (`stipendio`) = 80
- Dimensione RID = 4 byte
- Dimensione PID = 3 byte
- Percentuale di riempimento foglie = 98%
- Percentuale di riempimento nodi = 87.5%

Si ricordi che in un R-tree le foglie contengono coppie (chiave, lista di RID), mentre i nodi interni contengono coppie (MBB, PID) e che per la memorizzazione di un MBB richiede una coppia di valori per ciascun attributo.

Esercizio 2 (5 punti)

Date le relazioni con schema:

```
Esami(matricola, codice, data, voto)
Studenti(matricola, nome, cognome, data, luogo, cdl)
```

si ottimizzi l'esecuzione della seguente interrogazione SQL:

```
SELECT S.nome, S.cognome, E.voto
FROM Studenti S, Esami E
WHERE S.matricola=E.matricola
  AND E.data >= 1-1-2011
  AND S.cdl in ('0937', '0234', '0049', '0050')
```

tenendo conto che dai cataloghi della base di dati risulta:

- Numero di tuple `Esami` = 240K
- Numero di pagine `Esami` = 1K
- Numero di tuple `Studenti` = 40K
- Numero di pagine `Studenti` = 5K
- Indice clustered su `E.data`: numero foglie = 25, numero anni = 10
- Indice unclustered (TID disordinate) su (`E.matricola`, `E.codice`): numero foglie = 20, numero corsi = 100
- Indice unclustered su `S.matricola`: numero foglie = 200
- Indice clustered su `cdl`: numero foglie = 10, numero corsi di laurea = 50

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 = 1000:

R	Φ(R, P)
100	95.20785289
200	181.3511705
300	259.2929678
400	329.814094
500	393.6210551
600	451.3530925
700	503.5885866
800	550.8508514
900	593.6133775
1000	632.3045752

R	Φ(R, P)
1100	667.3120671
1200	698.9865709
1300	727.6454132
1400	753.5757086
1500	777.0372363
1600	798.2650423
1700	817.4717945
1800	834.849913
1900	850.5734982
2000	864.8000746

R	Φ(R, P)
2100	877.6721692
2200	889.3187393
2300	899.8564645
2400	909.3909155
2500	918.0176119
2600	925.822979
2700	932.8852139
2800	939.2750686
2900	945.0565589
3000	950.287606

Esercizio 3 (5 punti)

Si illustri il funzionamento dell'ottimizzatore delle interrogazioni di System R, evidenziandone le peculiarità e le possibili cause di sub-ottimalità della soluzione.

Esercizio 4 (3 punti)

Si illustri quale sia l'effetto di incrementare/decrementare la dimensione della pagina dati su una (o più) struttura hash a scelta, discutendone in particolare l'impatto sui costi di inserimento e ricerca.

Soluzione Esercizio 1

Numero di coppie (`annodinascita, stipendio`) = 25 x 80 = 2000, mediamente ci sono 20K/2K = 10 tuple per ogni valore di chiave.
Dimensione di ogni record (foglia) = 8 + 8 + 10 x 4 = **56B**
Dimensione "reale" foglia = (4096 – 96) x 0.98 = **3920B**
Numero di record per foglia = 3920/56 = **70**
Numero di foglie = 2K/70 = **29**
Dimensione di ogni record (nodo interno) = 3 + 2 x (8 + 8) = **35B**
Dimensione "reale" nodo = (4096 – 96) x 0.875 = **3500B**
Numero di record per nodo interno = 3500/35 = **100**
Numero nodi livello 1 = 29/100 = **1**
L'R-tree corrispondente si compone quindi di 2 livelli per un totale di 1 nodi interni e 29 foglie.

Soluzione Esercizio 2

Selettività dei predicati:
Predicato `E.data` = 1/10 = **0.1**
Predicato `S.cdl` = 1/50 = **0.02** per ogni valore di `cdl`
Predicato di join = 1/40K (chiave esterna)

Accesso a S:
Costo scan sequenziale = **5000**
Costo indice su `cdl`: 4 x (NL x 0.02 + NP x 0.02) = 4 x (10 x 0.02 + 5000 x 0.02) = 4 x (1 + 100) = **404**
Costo indice su `matricola`: 1 + 1 = **2**
Numero tuple residue = 4 x NT x 0.02 = **3200**

Accesso a E:
Costo scan sequenziale = **1000**
Costo indice su `data`: NL x 0.1 + NP x 0.1 = 25 x 0.1 + 1000 x 0.1 = 3 + 100 = = **103**
Costo indice su (`E.matricola`, `E.codice`): NL/40K + NT/40K = 1K/40K + 240K/40K = 1 + 6 = **7**
Numero tuple residue = NT x 0.1 = **24000**

Costi di join:
`S` esterna: costo = costo indice su `cdl` + 3200 x costo indice `matricola` = 404 + 2000 x 7 = **14404**
`E` esterna: costo = costo indice su `data` + 24000 x costo indice su `matricola` = 103 + 24000 x 2 = **48103**
Il numero di risultati dell'interrogazione è 240K x 0.1 x 0.08 = **1920**