Tecnologie delle Basi di Dati M

Appello del 18/2/2013

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 = 5M
- Numero di pagine = 11.2K
- Dimensione `cognome` = dimensione `nome` = 15 byte
- Dimensione RID = 6 byte
- Dimensione PID = 2 byte
- Percentuale di riempimento foglie indice = 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 P.nome LIKE 'P%'
  AND R.luogo IN ('Milano', 'Bologna', 'Torino')
```

tenendo conto che dai cataloghi della base di dati risulta:

- Numero di tuple `Personale` = 40K
- Numero di pagine `Personale` = 2K
- Indice clustered su `luogo`: numero foglie = 100, numero chiavi = 50
- Indice unclustered (TID ordinate) su `nome`: numero foglie = 1K, i valori iniziano con una consonante dell'alfabeto italiano.
- Indice unclustered (TID ordinate) su `responsabile`: numero foglie = 4K, numero di responsabili = 100.
- 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, 2000)
100	97.56
200	190.37
300	278.65
400	362.62
500	442.50
600	518.47
700	590.75
800	659.49
900	724.89
1000	787.09

R	Φ(R, 2000)
1100	846.26
1200	902.54
1300	956.08
1400	1007.00
1500	1055.44
1600	1101.52
1700	1145.35
1800	1187.04
1900	1226.70
2000	1264.43

R	Φ(R, 2000)
2100	1300.31
2200	1334.44
2300	1366.91
2400	1397.79
2500	1427.17
2600	1455.11
2700	1481.69
2800	1506.98
2900	1531.03
3000	1553.91

Esercizio 3 (5 punti)

Si confrontino i protocolli 2PL e Strict 2PL, specificando in particolare i motivi per cui il secondo è da preferirsi rispetto al primo.

Esercizio 4 (3 punti)

Supponendo di dovere ottimizzare un'interrogazione su due relazioni in cui il predicato di join sia del tipo `A.attr < B.attr`, si indichi come dovrebbero essere modificati i vari algoritmi di join conosciuti, precisando quali di essi non risultino applicabili/efficienti.

Soluzione Esercizio 1

Dimensionamento indice denso:
Dimensione di ogni record (foglia) = 6 + 15 + 15 = **36B**
Dimensione "reale" foglia/nodo = (4096 – 96) x 0.90 = **3600B**
Numero di record per foglia = 3600/360 = **100**
Numero di foglie = 5M/100 = **50000**
Dimensione di ogni record (nodo interno) = 2 + 15 + 15 = **32B**
Numero di record per nodo interno = 4000/32 = **125**
Numero nodi livello 1 = 50000/125 = **400**
Numero nodi livello 2 = 400/125 = **4**
Numero nodi livello 3 = 4/125 = **1**
Il B^+-tree corrispondente si compone quindi di 4 livelli per un totale di 405 nodi interni e 50000 foglie.
Dimensionamento indice sparso:
Dimensione di ogni record (foglia/nodo) = 2 + 15 + 15 = **32B**
Numero di record per foglia/nodo = 3600/32 = **112**
Numero di foglie = 11.2K/112 = **100**
Numero nodi livello 1 = 100/125 = **1**
Il B^+-tree corrispondente si compone quindi di 2 livelli per un totale di 1 nodo radice e 100 foglie.

Soluzione Esercizio 2

Selettività dei predicati:
Predicato `P.nome LIKE 'P%'` = 1/16 = **0.0625**
Predicato `R.luogo IN ('Milano', 'Bologna', 'Torino')` = 1/50 = **0.02** per ogni valore di `luogo`

Accesso a P:
Costo scan sequenziale = **2000**
Costo indice su `nome`: NL x 0.0625 + Φ(NT x 0.0625, NP) = 1K x 0.0625 + Φ(40K x 0.0625, 2K) = 63 + Φ(2500, 2000)) = 63 + 1428 = **1491**
Costo indice su `responsabile`: NL x 0.01 + Φ(NT x 0.01, NP) = 4K x 0.01 + Φ(40K x 0.01, 2K) = 40 + Φ(400, 2000)) = 40 + 363 = **403**
Numero tuple residue = NT x 0.0625 = **2500**

Accesso a R:
Costo scan sequenziale = **2000**
Costo indice su `luogo`: 3 x (NL x 0.02 + NP x 0.02) = 3 x (100 x 0.02 + 2000 x 0.02) = 3 x (2 + 40) = **126**
Costo indice su `matricola`: 1 + 1 = **2**
Numero tuple residue = 3 x NT x 0.02 = **2400**

Costi di join:
`P` esterna: costo = costo indice su `nome` + 2500 x costo indice `matricola` = 1491 + 2500 x 2 = **6491**
`R` esterna: costo = costo indice su `luogo` + 2400 x costo indice su `responsabile` = 126 + 2400 x 403 = **967326**
Il numero di risultati dell'interrogazione è 40K x 0.0625 x 3 x 0.02 = **150**