

Tecnologie e architetture per la gestione dei dati

Prova parziale — 5 aprile 2024

Tempo a disposizione: un'ora e 20 minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%)

Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 10, 9, 8, 7, 6, 13, 14, 15, 30, 32, 5. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Domanda 2 (25%)

Si considerino due relazioni $R_1(\underline{A}, C, D, E)$, $R_2(\underline{A}, F, G, H)$, in cui gli attributi hanno tutti la stessa dimensione $L = 2\text{Byte}$, molto più piccola della dimensione del blocco pari a $B = 8000\text{ Byte}$. Si supponga che le relazioni abbiano entrambe $N = 4.000.000$ ennuple, con gli stessi valori su A , **che sono interi consecutivi da 1 a 4.000.000** e che le operazioni più frequenti su di essa siano le seguenti:

- o_1 selezione di 100 di ennuple del join di R_1 e R_2 (selezionate sulla base di valori consecutivi di A , ad esempio da 1.000.001 a 1.000.100), con frequenza $f_1 = 10.000$;
- o_2 scansione dell'intera relazione R_1 , con frequenza $f_2 = 1$

Valutare le due seguenti alternative di memorizzazione, calcolando il costo complessivo (riportare la formula che indica il numero di accessi nell'unità di tempo, in base alle variabili sopra citate):

- (i) memorizzazione separata delle due relazioni, entrambe ordinate su A e con indice primario su A con profondità $p = 4$, con 2 livelli mediamente disponibili nel buffer.

- (ii) memorizzazione in un cluster delle due relazioni pure entrambe ordinate su A e con indice primario su A con profondità sempre $p = 4$, con 2 livelli mediamente disponibili nel buffer.

In conclusione, conviene quindi la memorizzazione in un cluster? (Sì o No)

Tecnologie e architetture per la gestione dei dati — 5 aprile 2024

Ripetere la valutazione effettuata alla pagina precedente nel caso in cui le due frequenze sono invece $f_1 = 10$ e $f_2 = 10$

(i) memorizzazione separata delle due relazioni ...

(ii) memorizzazione in un cluster delle due relazioni ...

In conclusione, conviene quindi la memorizzazione in un cluster? (Sì o No)

Domanda 3 (30%)

Si considerino un sistema con blocchi di $B = 4000$ byte e una relazione $R(\text{Matricola}, \text{CodiceFiscale}, \text{Cognome}, \dots)$ di cardinalità pari circa a $L = 10.000.000$, con ennuple di $l = 40$ byte. Il campo *Matricola* è chiave di tipo intero e il campo *CodiceFiscale* è di tipo stringa e anch'esso chiave — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

- strutture primarie di due tipi: (a) disordinate, (b) hash
- indici secondari di tipo B-tree (anche più di uno sulla stessa relazione)

Considerare un carico applicativo con le seguenti operazioni

1. inserimento di una ennupla con numero di matricola diverso da tutti quelli già presenti (succede sempre così ma viene comunque verificato il vincolo); anche il codice fiscale è diverso da quelli esistenti (ma anche in questo caso il vincolo va verificato), con frequenza giornaliera $f_1 = 10.000$
2. ricerca di una ennupla sulla base del valore (completo) della matricola, con frequenza giornaliera $f_2 = 10.000$;
3. ricerca di una ennupla sulla base del valore (completo) del codice fiscale, con frequenza giornaliera $f_3 = 10$;
4. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) del cognome, con frequenza giornaliera $f_4 = 10$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s_4 = 100$ ennuple;
5. scansione dell'intera relazione, ordinata sulla base del valore del codice fiscale, con frequenza giornaliera $f_5 = 1$;

Progettare l'organizzazione fisica della relazione, (i) scegliendo la struttura primaria fra le varie possibilità e (ii) individuando gli eventuali indici. Ragionare in termini di numero di accessi a memoria secondaria, assumendo che (1) gli indici secondari abbiano profondità $p = 4$, (2) il buffer disponibile abbia (2a) dimensione minore del numero di blocchi del file e maggiore della radice quadrata di tale numero e (2b) permetta di mantenere stabilmente in memoria due livelli di indice, sia per i primari sia per i secondari. Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere commentando prima di tutto, qui sotto, i criteri sulla base dei quali sono state scelte le alternative ritenute migliori e poi alla pagina successiva specificare i costi, in forma sia simbolica sia numerica.

Tecnologie e architetture per la gestione dei dati — 5 aprile 2024

| | Alternativa 1 | Alternativa 2 | Alternativa 3 (eventuale) |
|-----------------------|---------------|---------------|---------------------------|
| Descrizione struttura | | | |
| Costo operazione 1 | | | |
| Costo operazione 2 | | | |
| Costo operazione 3 | | | |
| Costo operazione 4 | | | |
| Costo operazione 5 | | | |
| Costo totale | | | |

Domanda 4 (25%)

1. Considerare un sistema con blocchi di dimensione $B = 4\text{KB}$ approssimabile come 4.000 e una relazione $R(\underline{A}, C, D)$ contenente circa $L = 400.000$ ennuple di $r = 10$ Byte ciascuna, con $v_C = 400$ valori diversi per l'attributo C , compresi fra 1 e 400 e uniformemente distribuiti (quindi per ogni valore di C ci sono circa $L/v_C = 1000$ ennuple con tale valore) e $v_D = 200$ valori diversi per l'attributo D , compresi fra 1 e 200 e pure uniformemente distribuiti (quindi per ogni valore di D ci sono circa $L/v_D = 2000$ ennuple con tale valore).

La relazione ha un indice su C e uno su D . Supporre che (i) tali indici abbiano tre livelli e siano realizzati con riferimenti ai record, (ii) ciascun riferimento (nell'indice) occupi 10 Byte e (iii) di disporre di buffer sufficiente per memorizzare i riferimenti a 10.000 record.

In tale contesto, considerare la seguente interrogazione SQL

```
SELECT * FROM R WHERE C=15 AND D=3
```

Sulla base di quanto detto sopra, si può supporre che il risultato sia costituito in media da 5 ennuple, perché per ogni valore di C ci sono 1000 ennuple, con potenzialmente 200 valori diversi di D .

Indicare brevemente quali potrebbero essere il piano di accesso e il costo per l'esecuzione dell'interrogazione.

2. Considerare ancora la relazione $R(\underline{A}, C, D)$, di cui alla pagina precedente, con però 40.000 valori diversi per l'attributo C e 4 valori diversi per l'attributo D . Indicare ancora piano di accesso e costo dell'interrogazione.

3. Considerare ancora la relazione $R(\underline{A}, C, D)$, con 20 valori diversi per l'attributo C e 10 valori diversi per l'attributo D . Indicare ancora piano di accesso e costo dell'interrogazione.

Tecnologie e architetture per la gestione dei dati

Prova parziale — 5 aprile 2024

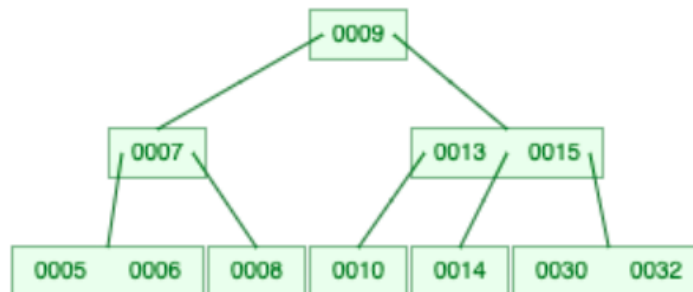
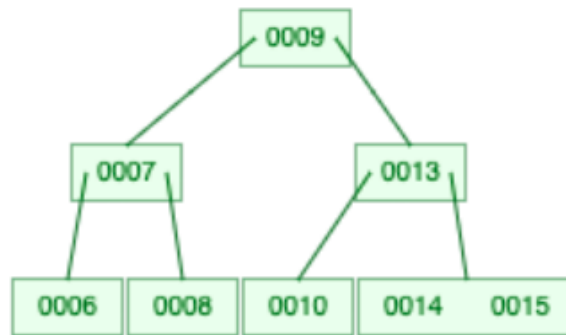
Cenni sulle soluzioni

Tempo a disposizione: un'ora e 20 minuti.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (20%)

Si consideri un B-tree con nodi intermedi che contengono due chiavi e tre puntatori e foglie con due chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 10, 9, 8, 7, 6, 13, 14, 15, 30, 32, 5. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.



Domanda 2 (25%)

Si considerino due relazioni $R_1(\underline{A}, C, D, E)$, $R_2(\underline{A}, F, G, H)$, in cui gli attributi hanno tutti la stessa dimensione $L = 2\text{Byte}$, molto più piccola della dimensione del blocco pari a $B = 8000\text{ Byte}$. Si supponga che le relazioni abbiano entrambe $N = 4.000.000$ ennuple, con gli stessi valori su A , **che sono interi consecutivi da 1 a 4.000.000** e che le operazioni più frequenti su di essa siano le seguenti:

- o_1 selezione di 100 di ennuple del join di R_1 e R_2 (selezionate sulla base di valori consecutivi di A , ad esempio da 1.000.001 a 1.000.100), con frequenza $f_1 = 10.000$;
- o_2 scansione dell'intera relazione R_1 , con frequenza $f_2 = 1$

Valutare le due seguenti alternative di memorizzazione, calcolando il costo complessivo (riportare la formula che indica il numero di accessi nell'unità di tempo, in base alle variabili sopra citate):

- (i) memorizzazione separata delle due relazioni, entrambe ordinate su A e con indice primario su A con profondità $p = 4$, con 2 livelli mediamente disponibili nel buffer.

| | |
|---|---|
| <p>Per o_1, le 100 ennuple sono, in ciascuno dei due casi, in un solo blocco (o in pochi casi in due, visto che il fattore di blocco è 1000; vanno utilizzati i due indici costo :</p> $c_1 = 2 + 1 + 2 + 1 = 6$ | <p>Per o_2, si debbono visitare tutti i blocchi di R_1:</p> $c_2 = \frac{N}{B/(4 \times L)} = \frac{4 \times N \times L}{B} = 4000$ |
| <p>costo complessivo:</p> $c_1 \times f_1 + c_2 \times f_2 = 6 \times 10.000 + 4000 \times 1 = 64.000$ | |

- (ii) memorizzazione in un cluster delle due relazioni pure entrambe ordinate su A e con indice primario su A con profondità sempre $p = 4$, con 2 livelli mediamente disponibili nel buffer.

| | |
|--|--|
| <p>costo unitario di o_1:</p> $c_1 = 3$ | <p>costo unitario di o_2:</p> $c_2 = \frac{N}{B/(8 \times L)} = \frac{8 \times N \times L}{B} = 8000$ |
| <p>costo complessivo:</p> $c_1 \times f_1 + c_2 \times f_2 = 3 \times 10.000 + 8000 \times 1 = 38.000$ | |

In conclusione, conviene quindi la memorizzazione in un cluster? (Sì o No)

SI

Tecnologie e architetture per la gestione dei dati — 5 aprile 2024

Ripetere la valutazione effettuata alla pagina precedente nel caso in cui le due frequenze sono invece $f_1 = 10$ e $f_2 = 10$

(i) memorizzazione separata delle due relazioni ...

cambia solo il costo complessivo:

$$c_1 \times f_1 + c_2 \times f_2 = 6 \times 10 + 4000 \times 10 = ca\ 40.000$$

(ii) memorizzazione in un cluster delle due relazioni ...

cambia solo il costo complessivo:

$$c_1 \times f_1 + c_2 \times f_2 = 3 \times 10 + 8000 \times 10 = ca\ 80.000$$

In conclusione, conviene quindi la memorizzazione in un cluster? (Sì o No)

NO

Domanda 3 (30%)

Si considerino un sistema con blocchi di $B = 4000$ byte e una relazione $R(\text{Matricola}, \text{CodiceFiscale}, \text{Cognome}, \dots)$ di cardinalità pari circa a $L = 10.000.000$, con ennuple di $l = 40$ byte. Il campo *Matricola* è chiave di tipo intero e il campo *CodiceFiscale* è di tipo stringa e anch'esso chiave — quindi la relazione ha due chiavi, ognuna delle quali, da sola, identifica univocamente le ennuple. Supporre che il sistema offra

- strutture primarie di due tipi: (a) disordinate, (b) hash
- indici secondari di tipo B-tree (anche più di uno sulla stessa relazione)

Considerare un carico applicativo con le seguenti operazioni

1. inserimento di una ennupla con numero di matricola diverso da tutti quelli già presenti (succede sempre così ma viene comunque verificato il vincolo); anche il codice fiscale è diverso da quelli esistenti (ma anche in questo caso il vincolo va verificato), con frequenza giornaliera $f_1 = 10.000$
2. ricerca di una ennupla sulla base del valore (completo) della matricola, con frequenza giornaliera $f_2 = 10.000$;
3. ricerca di una ennupla sulla base del valore (completo) del codice fiscale, con frequenza giornaliera $f_3 = 10$;
4. ricerca di una ennupla sulla base del valore parziale (una sottostringa iniziale) del cognome, con frequenza giornaliera $f_4 = 10$; supporre che il valore parziale sia poco selettivo e porti alla identificazione, in media, di $s_4 = 100$ ennuple;
5. scansione dell'intera relazione, ordinata sulla base del valore del codice fiscale, con frequenza giornaliera $f_5 = 1$;

Progettare l'organizzazione fisica della relazione, (i) scegliendo la struttura primaria fra le varie possibilità e (ii) individuando gli eventuali indici. Ragionare in termini di numero di accessi a memoria secondaria, assumendo che (1) gli indici secondari abbiano profondità $p = 4$, (2) il buffer disponibile abbia (2a) dimensione minore del numero di blocchi del file e maggiore della radice quadrata di tale numero e (2b) permetta di mantenere stabilmente in memoria due livelli di indice, sia per i primari sia per i secondari.

Proporre almeno due alternative (quelle che intuitivamente si ritengono migliori) e valutarne il costo. Rispondere commentando prima di tutto, qui sotto, i criteri sulla base dei quali sono state scelte le alternative ritenute migliori e poi alla pagina successiva specificare i costi, in forma sia simbolica sia numerica.

Osservazioni

- R ha un fattore di blocco B/l pari a 100 e quindi occupa $L_B = (L/(B/l)) = 100.000$ blocchi.
- L'operazione 5 è irrilevante, perché non è possibile mantenere l'ordinamento e quindi può essere ignorata, avendo lo stesso costo in tutte le alternative (3 volte il numero dei blocchi)
- vista la necessità di verificare le chiavi, serve un indice (o una struttura hash) su ciascuna delle chiavi; poiché l'operazione 2 è più frequente, meglio hash su matricola e indice su CF
- a parità del resto, una soluzione con hash su matricola probabilmente domina quella con hash su CF
- si può pensare ad un indice su cognome, che avrebbe costi aggiuntivi per l'operazione 1 (aggiornamento dell'indice) e benefici per l'operazione 4, è necessario valutare quantitativamente pro e contro
- si può pensare ad una soluzione con tre indici, che però è dominata (per via delle operazioni 1 e 2) da quella con hash su matricola e due indici

Quindi, sono comunque opportuni

- hash su matricola
- indice su codice fiscale

Va valutato l'indice su cognome

Nota Le due soluzioni indicate come “dominate” possono essere migliori di quella senza indice su cognome, ma possono non essere calcolate perché peggiori di quella con hash e due indici

Tecnologie e architetture per la gestione dei dati — 5 aprile 2024

| | Alternativa 1 | Alternativa 2 | Alternativa 3 (eventuale) |
|----------------|--|---|---------------------------|
| Descr. strutt. | Hash su Matricola e indice secondario su CodiceFiscale | Idem più indice secondario su Cognome | |
| Costo Op. 1 | visita e aggiornamento dell' indice $p_1 - 2 + 1$ e accesso hash al blocco 1 e scrittura del blocco con il nuovo record 1 totale 5 pesato $5 \times 10.000 =$ 50.000 | idem più visita e aggiornamento del terzo indice totale 8 pesato $8 \times 10.000 =$ 80.000 | |
| Costo Op. 2 | 1=c pesato 10.000 | idem 10.000 | |
| Costo Op. 3 | $p_2-2+1=3$: visita indice (p_2-2) e 1 record; totale 3, pesato 30 | idem 30 | |
| Costo Op. 4 | Serve scansione: $L/(B/c)=100.000$ pesato 1.000.000 | $p_2-2+10=ca.100$: visita indice (p_2-2) e 100 record; totale ca 100, pesato 1000 | |
| Costo Op. 5 | $3 \times L/(B/c)=300.000$ totale 300.000 | Idem 300.000 | |
| Tot | ca 1.360.000 | ca 390.000 | |

Domanda 4 (25%)

1. Considerare un sistema con blocchi di dimensione $B = 4\text{KB}$ approssimabile come 4.000 e una relazione $R(\underline{A}, C, D)$ contenente circa $L = 400.000$ ennuple di $r = 10$ Byte ciascuna, con $v_C = 400$ valori diversi per l'attributo C , compresi fra 1 e 400 e uniformemente distribuiti (quindi per ogni valore di C ci sono circa $L/v_C = 1000$ ennuple con tale valore) e $v_D = 200$ valori diversi per l'attributo D , compresi fra 1 e 200 e pure uniformemente distribuiti (quindi per ogni valore di D ci sono circa $L/v_D = 2000$ ennuple con tale valore).

La relazione ha un indice su C e uno su D . Supporre che (i) tali indici abbiano tre livelli e siano realizzati con riferimenti ai record, (ii) ciascun riferimento (nell'indice) occupi 10 Byte e (iii) di disporre di buffer sufficiente per memorizzare i riferimenti a 10.000 record.

In tale contesto, considerare la seguente interrogazione SQL

```
SELECT * FROM R WHERE C=15 AND D=3
```

Sulla base di quanto detto sopra, si può supporre che il risultato sia costituito in media da 5 ennuple, perché per ogni valore di C ci sono 1000 ennuple, con potenzialmente 200 valori diversi di D .

Indicare brevemente quali potrebbero essere il piano di accesso e il costo per l'esecuzione dell'interrogazione.

Sulla base delle informazioni fornite, ci sono 1000 ennuple (che potrebbero trovarsi in molti blocchi diversi) con valore $C=15$ e 2000 ennuple con valore $D=3$.

Però il risultato è molto piccolo, con solo 5 ennuple.

Il modo migliore per limitare gli accessi si può ottenere lavorando solo sugli indici e facendo l'intersezione di riferimenti presi dai due indici: prendendo dall'indice su C i riferimenti alle 1000 ennuple di interesse, il che può richiedere accesso a 1000×10 Byte consecutivi sull'indice, corrispondenti a 2 o 3 blocchi, più un altro accesso (o due) per scendere nell'indice e la stessa cosa sull'indice su D (2000 riferimenti, pari a 3 o 4 blocchi) più un altro accesso (o due) per l'indice. Infine si dovrà accedere ai cinque record. In totale circa 15 o 16 accessi.

Qualunque altro metodo (ad esempio quelli usati nelle due operazioni successive) richiede molti più accessi

2. Considerare ancora la relazione $R(\underline{A}, C, D)$, di cui alla pagina precedente, con però 40.000 valori diversi per l'attributo C e 4 valori diversi per l'attributo D . Indicare ancora piano di accesso e costo dell'interrogazione.

In questo caso, per la seconda relazione, i riferimenti sarebbero moltissimi e l'intersezione non conviene. Però la condizione su C è molto selettiva e restituisce 10 record che possono essere letti, per vedere se il valore di D è quello di interesse. In totale servono due o tre accessi per l'indice più dieci per i record per un totale di 12 o 13

3. Considerare ancora la relazione $R(\underline{A}, C, D)$, con 20 valori diversi per l'attributo C e 10 valori diversi per l'attributo D . Indicare ancora piano di accesso e costo dell'interrogazione.

In questo caso le condizioni sono poco selettive e il risultato contiene circa 2000 tuple, sparpagliate nei vari blocchi. Poiché la relazione contiene circa 1000 blocchi, conviene procedere con una scansione sequenziale, costo appunto 1000.