

Tecnologia delle basi di dati (ex Basi di dati, primo modulo)

16 luglio 2007 — Compito A

Tempo a disposizione: 2 ore e 15 minuti. **Nota:** è richiesta una “bella copia” comprensibile e ordinata.

Domanda 1 (25%) Sia data una relazione $R(A, B, C)$ contenente $L = 10.000.000$ ennuple di $r = 20$ byte ciascuna, di cui $a = 4$ per la chiave A , che contiene valori interi consecutivi, da 1 a 10.000.000. In ciascuno dei seguenti casi:

- (a) indice primario (sparso) su A realizzato con B+-tree;
- (b) indice secondario su A realizzato con B+-tree;
- (c) struttura primaria hash su A .

indicare l'algoritmo presumibilmente preferibile e il costo per

- l'accesso ai record con valore della chiave A compreso fra 1.000 e 3.000,
- l'accesso ai record con valore della chiave A compreso fra 100.000 e 300.000

Supporre che i blocchi abbiano dimensione 2KB, approssimabile come 2.000, e che i puntatori ai blocchi abbiano lunghezza $p = 4$; ignorare la presenza di buffer.

Possibile soluzione

- (a1) costo $LIV_1 + b = 23$, dove LIV_1 è la profondità dell'indice, pari a $\lceil \log_{F'} L/(B/r) \rceil = 3$, dove $F' = 2/3F$ (supponendo un riempimento medio del 66%) e $F = B/(a + p)$ è il fattore di blocco massimo dell'indice e $b = 2.000/(B/r) = 20$ è il numero di blocchi occupati dai 2.000 record di interesse;
- (a2) simile, con $b = 2.000$
- (b1) costo $LIV_2 + n(1 + 1/F') = 2.000$ circa, dove LIV_2 è la profondità dell'indice, pari a $\lceil \log_{F'} L \rceil = 4$, e $n = 2.000$ è il numero di record di interesse, che vanno visitati tutti; inoltre vi sono $n/F' = 12$ foglie dell'indice da visitare per trovare i riferimenti ad esse — trascuriamo il numero;
- (b1) utilizzando l'indice, si avrebbe un costo (come nel caso precedente) dell'ordine di n , ma in questo caso $n = 200.000$, maggiore della dimensione in blocchi del file, pari a $L/(B/r) = 100.000$; conviene quindi una scansione sequenziale
- (c1) costo poco maggiore di n (con n accessi hash diversi);
- (c2) utilizzando l'accesso hash si avrebbe costo poco maggiore di n e quindi conviene l'accesso sequenziale

Domanda 2 (20%) Il *semijoin* è un'operazione simile al join, in cui del secondo operando interessano solo gli attributi di join. In concreto, se il join su $A_1 = A_2$ di $r_1(X_1)$ e $r_2(X_2)$ (con $A_1 \in X_1$, $A_2 \in X_2$ e $X_1 \cap X_2 = \emptyset$) è definito come

$$r_1 \text{ JOIN}_{A_1=A_2} r_2 = \{ t \text{ su } X_1 X_2 \mid \text{esistono } t_1 \in r_1 \text{ e } t_2 \in r_2 \text{ con } t[X_1] = t_1, t[X_2] = t_2 \text{ e } t_1[A_1] = t_2[A_2] \}$$

il semijoin corrispondente è definito come

$$r_1 \text{ SEMIJOIN}_{A_1=A_2} r_2 = \{ t_1 \mid t_1 \in r_1 \text{ ed esiste } t_2 \in r_2 \text{ con } t_1[A_1] = t_2[A_2] \}$$

Ad esempio:

R_1				R_2		$R_1 \text{ SEMIJOIN}_{D=G} R_2$			
A	B	C	D	G	E	A	B	C	D
1	1	1	1	1	1	1	1	1	1
2	2	2	1	3	2	2	2	2	1
3	2	2	2						

Mostrare algoritmi per l'esecuzione del semijoin, come modifica degli algoritmi noti per il join, indicandone il costo (e sottolineando i casi in cui tale costo è diverso da quello del join).

Inoltre, specificare quali possono essere i vantaggi dell'uso del semijoin in contesti distribuiti (con le due relazioni memorizzate in nodi diversi), anche come passo preliminare per l'esecuzione di join.

Possibile soluzione

Gli algoritmi possono fermarsi alla verifica della presenza della ennupla del secondo argomento, senza accedere alle relative ennuple. Ciò diventa significativo quando si utilizza un indice (ad esempio in un nested loop con R_2 interna e indice su A_2 oppure in un merge scan che utilizzi un indice A_2 su R_2).

In un contesto distribuito, il semijoin può essere utile per evitare di trasferire un'intera relazione; in pratica si trasferiscono solo i valori degli attributi di join, realizzando quindi un semijoin, per poi trasferire solo le ennuple che certamente contribuiscono al join.

Domanda 3 (20%) Si supponga che Napoleone abbia fatto utilizzare il commit a due fasi per organizzare le attività in battaglia. Si consideri il seguente scenario. Esistono:

- (a) generali che possono coordinare azioni
- (b) reparti di riserva disponibili per azioni
- (c) messaggeri utilizzati dai generali e dai reparti di riserva per comunicare gli uni con gli altri

In particolare, il 2PC viene utilizzato dai generali per organizzare azioni che coinvolgono (*contemporaneamente, ad un certo orario*) due o più reparti di riserva (ma ciascun reparto potrebbe ricevere richieste da più generali). Ad esempio, il generale Murat, alle ore 10, decide di voler organizzare un attacco alle ore 12 con il coinvolgimento del quarto squadrone di cavalleria e della prima batteria di artiglieria pesante.

Descrivere brevemente il protocollo in questo contesto, sottolineando le criticità, dovute al fatto che si tratta di una battaglia e quindi tutti i soggetti coinvolti (generali, comandanti dei reparti di riserva e messaggeri) possono essere colpiti. In particolare, individuare quali ipotesi sarebbero necessarie (anche se non sempre soddisfatte in una battaglia) per permettere l'utilizzo del protocollo.

Possibile soluzione

- protocollo: un generale invia richieste a due o più reparti, se tutti rispondono affermativamente (perché hanno ricevuto il messaggio, sono liberi e riescono a far arrivare il messaggio di risposta) il generale comunica l'effettivo svolgimento dell'attacco, chiedendo conferma e continua a richiedere conferma finché non la riceve; ciascun reparto, se dichiara la propria disponibilità, non può poi ritirarla
- criticità:
 - ogni reparto deve avere la possibilità di ricordare messaggi inviati e ricevuti e quindi (ipotesi fondamentale) deve esistere un “registro” dei messaggi che possa essere consultato da un comandante che subentri qualora quello in carica venga colpito; l'ipotesi fondamentale è che il registro rimanga sempre integro (cosa ovviamente non scontata nel contesto in questione);
 - il requisito del rispetto dei tempi (che nel 2PC non c'è) non può essere garantito: la ripetizione dei messaggi nella seconda fase potrebbe non essere sufficiente (i reparti potrebbero ricevere la conferma in ritardo)

Domanda 4 (20%) Si consideri la base di dati seguente, relativa alla segreteria studenti di una università:

- Studenti(Matricola, Cognome, Nome, Anno) dove Anno indica l'anno di corso cui lo studente è iscritto (1 per il primo anno, 2 per il secondo, e così via);
- Corsi(Codice, Nome);
- Esami(Studente, Corso, Data, Voto), con vincoli di riferimento verso Studenti e verso Corsi.

Con riferimento a tale base di dati, progettare uno o più schemi dimensionali (specificando per ciascuno fatti, misure e dimensioni e mostrando qualche piccolo esempio delle relative tabelle con valori effettivi) che permettano di rispondere facilmente ad interrogazioni quali ad esempio (la lista non ha pretesa di essere esaustiva):

- calcolare il numero di studenti che hanno superato l'esame di un certo corso in un certo intervallo di tempo (specificato con giorno iniziale e giorno finale) e la media dei voti riportati;
- calcolare la distribuzione dei voti degli esami (di un corso o di tutti i corsi) in un certo intervallo di tempo (ad esempio, quanti hanno superato l'esame con 18, quanti con 19, etc., oppure quanti con voto compreso fra 18 e 20, quanti fra 21 e 23, etc.)
- calcolare la distribuzione in anni di corso degli studenti che hanno superato un certo esame (quanti al primo anno, quanti al secondo, e così via);

Possibile soluzione

Schema dimensionale

- FattiEsami(KData, KCorso, KVoto, KAnnoDiCorso, Voto, NumeroEsami)
- Corsi(KCorso, Codice, Nome)
- Voti(KVoto, Voto, FasciaDueVoti, FasciaTreVoti, ...)
- AnnoDiCorso(KAnnoDiCorso, Anno ...)
- Data(KData, ...)

Commenti:

- sono indicate chiavi ad hoc per le dimensioni
- Una soluzione alternativa poteva prevedere due schemi dimensionali, uno con i voti come dimensione e l'altro con il voto come misura (mentre quello mostrato li ha entrambi); le altre dimensioni sarebbero state comuni ai due schemi

Domanda 5 (15%) Si supponga che esista una misura di lunghezza non decimale, ad esempio il *pie*, suddiviso in dodici *pollici*. Spiegare perché un sistema di basi di dati a oggetti (object-oriented o object-relational) può essere, per la gestione di valori di tale misura, più efficace di un tradizionale sistema relazionale. Descrivere, anche informalmente, una possibile definizione per il tipo e per alcune funzioni associate, quali la somma e la differenza.

Possibile soluzione

Un modello a oggetti permette di definire un tipo con le operazioni associate. Nell'esempio, si creerà il tipo, probabilmente composto di due campi, con operazioni di somma e differenza che gestiscano opportunamente il riporto. A titolo di curiosità (ma non era richiesto all'esame) in DB2, si potrebbero avere le definizioni seguenti

```
CREATE TYPE PIEDI_E_POLLICI AS (  
  PIEDI INT,  
  POLLICI INT  
)  
MODE DB2SQL  
  METHOD SUMPIEDI_E_POLLICI (P1 PIEDI_E_POLLICI)  
  RETURNS PIEDI_E_POLLICI  
  LANGUAGE SQL  
  DETERMINISTIC  
  CONTAINS SQL  
  NO EXTERNAL ACTION  
  ;  
CREATE METHOD SUMPIEDI_E_POLLICI (P1 PIEDI_E_POLLICI )  
  FOR PIEDI_E_POLLICI  
  RETURN PIEDI_E_POLLICI (  
    ..PIEDI(P1..PIEDI + SELF..PIEDI+ cast((P1..POLLICI+self..POLLICI)/12 as integer))  
    ..POLLICI(MOD(P1..POLLICI+self..POLLICI,12))
```