

Basi di dati II

Esame — 29 settembre 2014

Rispondere su questo fascicolo.

Tempo a disposizione: due ore.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (15%)

Come dovrebbe essere noto, gli algoritmi che sfruttano i buffer per l'esecuzione dei join richiedono un numero di passate i che è, all'incirca, il più piccolo valore per cui il numero P di buffer disponibili è maggiore della radice i -esima del numero di blocchi dei file coinvolti.

1. Spiegare intuitivamente l'affermazione sopra riportata (ad esempio con riferimento al mergesort)

2. Spiegare perchè, con riferimento al mergesort, il calcolo in questione è abbastanza preciso, mentre, con riferimento all'hash-join, va considerato approssimato e, di conseguenza, può essere talvolta necessaria una passata aggiuntiva

Domanda 2 (15%)

Si supponga si abbiano documenti che contengono informazioni su opere teatrali e autori, come il seguente:

```
<opere teatrali>
  <opera>
    <titolo>Riccardo III</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
  </opera>
  <opera>
    <titolo>Amleto</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
  </opera>
  <opera>
    <titolo>La locandiera</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
  </opera>
  <opera>
    <titolo>La tempesta</titolo> <autore>Shakespeare</autore> <tipo>commedia</tipo>
  </opera>
  <opera>
    <titolo>I rusteghi</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
  </opera>
</opere teatrali>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli (mostrare solo le stringhe, non i nodi XML) delle opere il cui autore è Goldoni e che sono commedie

In XQuery, produrre un documento che contiene, per ogni autore che abbia scritto almeno una commedia, la lista di tali commedie. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<autore>
  Goldoni
    La locandiera
    I rusteghi
</autore>
<autore>
  Shakespeare
    La tempesta
</autore>
```

Domanda 3 (15%)

Considerare una tabella R contenente $N = 1.600.000$ ennuple, con le seguenti ipotesi

- R è definita su due campi, un campo chiave A di lunghezza $a = 4$ byte e un altro campo non chiave B di lunghezza $b = 14$ byte;
- la struttura fisica utilizzata per R è heap con memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi, ne servano 2 ulteriori per la memorizzazione);
- su R è definito un indice secondario $B+$ -tree sul campo chiave A , con riferimenti di $p = 4$ byte (e quindi si può assumere, contando 2 byte di servizio, che un record dell'indice occupi $l_i = 10$ byte).
- il sistema utilizza blocchi di dimensione $D = 4$ Kbyte (approssimabili a 4000).

Supporre che, in tale relazione vengano inserite ulteriori $T = 50.000$ ennuple, con valori della chiave tutti diversi fra loro e da quelli già nella relazione (quindi il sistema verifica il soddisfacimento del vincolo di chiave e ammette tutte le operazioni). Rispondere alle domande seguenti, indicando formule e valori numerici:

Indicare il numero di blocchi occupati da R all'inizio, supponendo che in ciascun blocco tutto lo spazio sia utilizzato.

Indicare il numero approssimativo di blocchi occupati dall'indice.

Indicare il numero di scritture in memoria secondaria per gli inserimenti, supponendo che i record di log abbiano una lunghezza pari a circa il triplo di quella dei record del file, con riferimento ad un programma che utilizzi una transazione separata per ciascun inserimento

- numero di scritture di pagine di log:

- numero di scritture di pagine della relazione:

- numero di scritture di pagine dell'indice (considerando solo le foglie):

Come nel caso precedente, ma con riferimento ad un programma che utilizzi un'unica transazione per tutti gli inserimenti (e supponendo che non vi siano altre transazioni attive)

- numero di scritture di pagine di log:

- numero di scritture di pagine della relazione:

- numero di scritture di pagine dell'indice (considerando solo le foglie):

Basi di dati II — 29 settembre 2014

Domanda 4 (15%) Considerare il seguente scenario in cui tre client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato).

client 1	client 2	client 3
begin read(x)		begin read(x)
x = x + 10 write(x) commit	begin read(x)	
	x = x + 20	read(x)
	write(x) commit	commit

Considerare uno scheduler con controllo di concorrenza basato su **2PL** e livello di isolamento **READ COMMITTED** su tutte le transazioni.

Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare, nell'ordine, le operazioni che vengono eseguite da ciascun client, specificando, per ciascuna, il valore che viene letto o scritto. In conclusione, dire se si verificano o meno anomalie.

client 1	client 2	client 3

Si verificano anomalie?

Basi di dati II — 29 settembre 2014

Domanda 5 (15%) Considerare il seguente scenario in cui tre client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato).

client 1	client 2	client 3
begin read(x)		
x = x + 10 write(x)	begin read(x)	
	x = x + 20 write(x) commit	begin read(x)
commit		read(x) commit

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** e livello di isolamento **SERIALIZABLE** su tutte le transazioni.

Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare, nell'ordine, le operazioni che vengono eseguite da ciascun client, specificando, per ciascuna, il valore che viene letto o scritto. In conclusione, dire se si verificano o meno anomalie.

client 1	client 2	client 3

Si verificano anomalie?

Indicare brevemente che cosa succede se invece la transazione sul client 3 ha livello di isolamento READ COMMITTED.

Domanda 6 (25%)

Considerare uno schema dimensionale relativo ai risultati delle elezioni politiche nelle varie regioni d'Italia che utilizzi, come tabella dei fatti e come una delle dimensioni, relazioni come le seguenti (KE denota la chiave della dimensione relativa alle elezioni, KP di quella relativa al partito e KR di quella relativa alla regione; i voti sono in migliaia):

<u>KE</u>	<u>KR</u>	<u>KP</u>	Voti	Percent	...
301	201	401	805	28,08	...
301	201	402	106	3,51	...
301	201	403	708	23,33	...
301	202	401	1454	27,31	...
302	201	403	75	7,8	...
...

<u>KP</u>	Sigla	Nome	...
401	PB	Partito Bello	...
403	PF	Partito Furbo	...
...

Con riferimento a questo contesto, considerare le esigenze di modifica seguenti. Per ciascuna, proporre una modifica allo schema e rispondere alle eventuali domande.

- i partiti cambiano nome nel tempo: per esempio, il “Partito Bello” potrebbe ad un certo punto diventare il “Partito Bello e Buono”; interessano selezioni e aggregazioni relative ai voti tanto con riferimento al nome del partito (al momento delle elezioni) quanto alla sua identità (un codice che viene introdotto allo scopo, ma non sempre viene utilizzato, perché alcuni analisti preferiscono fare riferimento al nome corrente del partito); le modifiche sono rare, ma è possibile che ci siano partiti con vari cambiamenti di nome; mostrare modifiche alle relazioni (una o entrambe) che permettano di soddisfare le esigenze sopra citate (mostrare anche i dati, con riferimento a quelli presenti negli esempi sopra, aggiungendo nuovi dati ragionevoli, che permettano di comprendere le modifiche).

- per ogni partito, interessa rappresentare anche il leader al momento dell'elezione, per supportare analisi sui risultati di ciascun leader; i leader cambiano nel tempo (e possono passare da un partito all'altro, sia pure raramente ...); è disponibile l'informazione relativa ai leader dei partiti nel tempo (per tutto il periodo, anche passato, di interesse).

Basi di dati II — 29 settembre 2014

- Nelle ultime tornate elettorali, è diventato rilevante il concetto di “coalizione”: ogni partito partecipa ad una coalizione (supponiamo per semplicità che un partito non coalizzato costituisca una coalizione da solo). Le coalizioni possono variare nel tempo. Proporre una ristrutturazione dello schema dimensionale e indicare se e come è possibile realizzare la modifica proposta anche con riferimento ad elezioni precedenti (indicare cioè quali dati debbono essere disponibili allo scopo nella staging area).

Basi di dati II

Esame — 29 settembre 2014

Cenni sulle soluzioni

Rispondere su questo fascicolo.
Tempo a disposizione: due ore.

Cognome _____ Nome _____ Matricola _____

Domanda 1 (15%)

Come dovrebbe essere noto, gli algoritmi che sfruttano i buffer per l'esecuzione dei join richiedono un numero di passate i che è, all'incirca, il più piccolo valore per cui il numero P di buffer disponibili è maggiore della radice i -esima del numero di blocchi dei file coinvolti.

1. Spiegare intuitivamente l'affermazione sopra riportata (ad esempio con riferimento al mergesort)

(N.B. Sia pure in modo sintetico, ci sia aspettava una risposta compiuta e chiara.)

Idee fondamentali:

Con P buffer disponibili, è possibile ordinare run di dimensione P e poi, con ogni passo di fusione, è possibile fondere P run (qualunque sia la loro lunghezza). Quindi con una passata (di ordinamento) si possono ordinare P blocchi, con due passate (la prima di ordinamento e la seconda di fusione) si ordinano P^2 blocchi, con n passate (una di ordinamento e $n - 1$ di fusione, si ordinano P^n blocchi. Quindi per ordinare B blocchi serve un numero di passate i tali che P^i sia maggiore di B .

2. Spiegare perchè, con riferimento al mergesort, il calcolo in questione è abbastanza preciso, mentre, con riferimento all'hash-join, va considerato approssimato e, di conseguenza, può essere talvolta necessaria una passata aggiuntiva

Idea fondamentale:

Perché la funzione hash (qualunque funzione hash) distribuisce in modo mediamente uniforme, ma non in parti esattamente uguali e quindi può succedere che non ci sia spazio per un run leggermente più grande.

Domanda 2 (15%)

Si supponga si abbiano documenti che contengono informazioni su opere teatrali e autori, come il seguente:

```
<opere teatrali>
  <opera>
    <titolo>Riccardo III</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
  </opera>
  <opera>
    <titolo>Amleto</titolo> <autore>Shakespeare</autore> <tipo>tragedia</tipo>
  </opera>
  <opera>
    <titolo>La locandiera</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
  </opera>
  <opera>
    <titolo>La tempesta</titolo> <autore>Shakespeare</autore> <tipo>commedia</tipo>
  </opera>
  <opera>
    <titolo>I rusteghi</titolo> <autore>Goldoni</autore> <tipo>commedia</tipo>
  </opera>
</opere teatrali>
```

Formulare le seguenti interrogazioni:

In XPath, trovare i titoli (mostrare solo le stringhe, non i nodi XML) delle opere il cui autore è Goldoni e che sono commedie

```
//opera[autore/text()="Goldoni"][tipo/text()="commedia"]/titolo/text()
```

In XQuery, produrre un documento che contiene, per ogni autore che abbia scritto almeno una commedia, la lista di tali commedie. Con riferimento al documento mostrato in precedenza, si vuole ottenere:

```
<autore>
  Goldoni
    La locandiera
    I rusteghi
</autore>
<autore>
  Shakespeare
    La tempesta
</autore>
```

Possibile risposta:

```
let $nl := "&#10;"
for $a in distinct-values(//opera[tipo="commedia"]/autore/text())
return
  <autore>
    {$a} {for $o in //opera[tipo="commedia"]
      where $o/autore = $a
      return ($nl, " ", $o/titolo/text())
    }
  </autore>
```

Domanda 3 (15%)

Considerare una tabella **R** contenente $N = 1.600.000$ ennuple, con le seguenti ipotesi

- **R** è definita su due campi, un campo chiave **A** di lunghezza $a = 4$ byte e un altro campo non chiave **B** di lunghezza $b = 14$ byte;
- la struttura fisica utilizzata per **R** è heap con memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi, ne servano 2 ulteriori per la memorizzazione);
- su **R** è definito un indice secondario $B+$ -tree sul campo chiave **A**, con riferimenti di $p = 4$ byte (e quindi si può assumere, contando 2 byte di servizio, che un record dell'indice occupi $l_i = 10$ byte).
- il sistema utilizza blocchi di dimensione $D = 4$ Kbyte (approssimabili a 4000).

Supporre che, in tale relazione vengano inserite ulteriori $T = 50.000$ ennuple, con valori della chiave tutti diversi fra loro e da quelli già nella relazione (quindi il sistema verifica il soddisfacimento del vincolo di chiave e ammette tutte le operazioni). Rispondere alle domande seguenti, indicando formule e valori numerici:

Indicare il numero di blocchi occupati da **R** all'inizio, supponendo che in ciascun blocco tutto lo spazio sia utilizzato.

$$N/f_R = 1.600.000/200 = 8.000$$

con f_R fattore di blocco per **R**: $f_R = D/(a + b + 2) = 4000/20 = 200$

Indicare il numero approssimativo di blocchi occupati dall'indice.

assumendo un riempimento di $2/3$, si può pensare che servano circa $3/2 \times N/f_i = 6.000$ foglie, e non più di una dozzina di altri nodi (f_i è il fattore di blocco (e quindi il fan-out) per l'indice: $f_i = D/(l_i) = 4000/10 = 400$)

Indicare il numero di scritture in memoria secondaria per gli inserimenti, supponendo che i record di log abbiano una lunghezza pari a circa il triplo di quella dei record del file, con riferimento ad un programma che utilizzi una transazione separata per ciascun inserimento

- numero di scritture di pagine di log:
almeno $T = 50.000$, una per transazione
- numero di scritture di pagine della relazione:
con una strategia undo-only, una per transazione, $T = 50.000$; altrimenti, probabilmente di meno, anche solo $T/f = 250$, una per blocco
- numero di scritture di pagine dell'indice (considerando solo le foglie):
con una strategia undo-only, una per transazione, $T = 50.000$; altrimenti, probabilmente di meno, anche solo una per foglia (quindi $3/2 \times N/f_i = 6.000$ come detto sopra)

Come nel caso precedente, ma con riferimento ad un programma che utilizzi un'unica transazione per tutti gli inserimenti (e supponendo che non vi siano altre transazioni attive)

- numero di scritture di pagine di log: dovrebbe essere possibile scrivere tutto il log al momento del commit e quindi (poiché i record sono grandi il triplo di quelli del file e per ogni record del file ce n'è uno dell'indice):
 $3 \times T/f_R + 3 \times T/f_i \approx 1100$
- numero di scritture di pagine della relazione:
in ogni caso si dovrebbe avere $T/f_R = 250$, una per blocco, o poco più
- numero di scritture di pagine dell'indice (considerando solo le foglie):
in ogni caso si dovrebbe avere una per foglia, o poco più

Basi di dati II — 29 settembre 2014

Domanda 4 (15%) Considerare il seguente scenario in cui tre client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato).

client 1	client 2	client 3
begin read(x)		begin read(x)
x = x + 10 write(x) commit	begin read(x)	
	x = x + 20	read(x)
	write(x) commit	commit

Considerare uno scheduler con controllo di concorrenza basato su **2PL** e livello di isolamento **READ COMMITTED** su tutte le transazioni.

Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare, nell'ordine, le operazioni che vengono eseguite da ciascun client, specificando, per ciascuna, il valore che viene letto o scritto. In conclusione, dire se si verificano o meno anomalie.

client 1	client 2	client 3
begin read(x) <i>legge 200</i>		begin read(x) <i>legge 200</i>
x = x + 10 write(x) <i>scrive 210</i> commit	begin read(x) <i>legge 200</i>	
	x = x + 20	read(x) <i>legge 210</i>
	write(x) <i>scrive 220</i> commit	commit

Si verificano anomalie?

perdita di aggiornamento fra per il client 1 e lettura inconsistente per il client 3

Basi di dati II — 29 settembre 2014

Domanda 5 (15%) Considerare il seguente scenario in cui tre client diversi inviano richieste ad un gestore del controllo di concorrenza. Ciascun client può inviare una richiesta solo dopo che è stata eseguita o rifiutata la precedente (se invece una richiesta viene bloccata da un lock, allora il client rimane inattivo fino alla concessione o allo scadere del timeout). Si supponga che, in caso di stallo, abortisca la transazione che ha avanzato la richiesta per prima. In caso di abort, si supponga che il client rilanci la stessa transazione (subito dopo l'esecuzione delle altre azioni in attesa sullo stesso dato).

client 1	client 2	client 3
begin read(x)		
x = x + 10 write(x)	begin read(x)	
	x = x + 20 write(x) commit	begin read(x)
commit		read(x) commit

Considerare uno scheduler con controllo di concorrenza basato su **Multiversioni** e livello di isolamento **SERIALIZABLE** su tutte le transazioni.

Mostrare il comportamento dello scheduler, supponendo che il valore iniziale dell'oggetto x sia 200. Indicare, nell'ordine, le operazioni che vengono eseguite da ciascun client, specificando, per ciascuna, il valore che viene letto o scritto. In conclusione, dire se si verificano o meno anomalie.

client 1	client 2	client 3
begin read(x) <i>legge 200</i>		
x = x + 10 write(x) <i>scrive 210</i>	begin read(x) <i>legge 200</i>	
	x = x + 20 xlock(x) <i>attesa</i>	begin read(x) <i>legge 200</i>
commit	abort begin read(x) <i>legge 210</i> write(x) <i>scrive 230</i> commit	
		read(x) <i>legge 200</i> commit

Si verificano anomalie?
nessuna

Indicare brevemente che cosa succede se invece la transazione sul client 3 ha livello di isolamento READ COMMITTED.

Risposta

Il cliente 3 legge valori in commit al momento e quindi la seconda lettura legge un valore diverso rispetto alla prima

Domanda 6 (25%)

Considerare uno schema dimensionale relativo ai risultati delle elezioni politiche nelle varie regioni d'Italia che utilizzi, come tabella dei fatti e come una delle dimensioni, relazioni come le seguenti (KE denota la chiave della dimensione relativa alle elezioni, KP di quella relativa al partito e KR di quella relativa alla regione; i voti sono in migliaia):

KE	KR	KP	Voti	Percent	...
301	201	401	805	28,08	...
301	201	402	106	3,51	...
301	201	403	708	23,33	...
301	202	401	1454	27,31	...
302	201	403	75	7,8	...
...

KP	Sigla	Nome	...
401	PB	Partito Bello	...
403	PF	Partito Furbo	...
...

Con riferimento a questo contesto, considerare le esigenze di modifica seguenti. Per ciascuna, proporre una modifica allo schema e rispondere alle eventuali domande.

- i partiti cambiano nome nel tempo: per esempio, il “Partito Bello” potrebbe ad un certo punto diventare il “Partito Bello e Buono”; interessano selezioni e aggregazioni relative ai voti tanto con riferimento al nome del partito (al momento delle elezioni) quanto alla sua identità (un codice che viene introdotto allo scopo, ma non sempre viene utilizzato, perché alcuni analisti preferiscono fare riferimento al nome corrente del partito); le modifiche sono rare, ma è possibile che ci siano partiti con vari cambiamenti di nome; mostrare modifiche alle relazioni (una o entrambe) che permettano di soddisfare le esigenze sopra citate (mostrare anche i dati, con riferimento a quelli presenti negli esempi sopra, aggiungendo nuovi dati ragionevoli, che permettano di comprendere le modifiche).

Per la dimensione può essere utile la tecnica della “slowly changing dimension” di tipo 2 (e contemporaneamente 3), con un nuovo elemento (quindi una ennupla nella relazione) per ogni modifica e, viste le specifiche, un codice aggiuntivo, che non cambia nel tempo, e due attributi per il nome, con quello attuale e quello “dell’epoca,” rispettivamente.

- per ogni partito, interessa rappresentare anche il leader al momento dell’elezione, per supportare analisi sui risultati di ciascun leader; i leader cambiano nel tempo (e possono passare da un partito all’altro, sia pure raramente ...); è disponibile l’informazione relativa ai leader dei partiti nel tempo (per tutto il periodo, anche passato, di interesse).

La tabella dei fatti può essere estesa aggiungendo una dimensione supplementare Leader, che dipende da Partito e Elezione e può quindi essere facilmente aggiunta anche ad una tabella dei fatti esistente (i dati, come detto, sono disponibili e le ennuple rimangono le stesse, con un attributo in più;):

Basi di dati II — 29 settembre 2014

- Nelle ultime tornate elettorali, è diventato rilevante il concetto di “coalizione”: ogni partito partecipa ad una coalizione (supponiamo per semplicità che un partito non coalizzato costituisca una coalizione da solo). Le coalizioni possono variare nel tempo. Proporre una ristrutturazione dello schema dimensionale e indicare se e come è possibile realizzare la modifica proposta anche con riferimento ad elezioni precedenti (indicare cioè quali dati debbono essere disponibili allo scopo nella staging area).

Ci sono diverse possibili soluzioni, quali quella di prevedere la coalizione nell'ambito della dimensione oppure una dimensione aggiuntiva