Basi di dati, primo modulo — Tecnologia delle basi di dati

18 giugno 2003 — Compito A — Tempo a disposizione: due ore

Domanda 1 (30%) Considerare le seguenti tre versioni del checkpoint e commentare brevemente le differenze in termini di prestazioni e di procedure di ripristino dopo i guasti

checkpoint, versione A

- si sospende l'accettazione di richieste di ogni tipo (scrittura, inserimenti, ..., commit, abort)
- si trasferiscono in memoria di massa (tramite force) tutte le pagine sporche relative a transazioni andate in commit
- si scrive sul log in modo sincrono (force) un record che indica il completamento del checkpoint, con gli identificatori delle transazioni in corso
- si riprende l'accettazione delle operazioni

checkpoint, versione B

- si sospende l'accettazione di nuove transazioni e si aspetta che le transazioni attive raggiungano la conclusione
- si trasferiscono in memoria di massa (tramite force) tutte le pagine sporche relative a transazioni andate in commit
- si scrive sul log in modo sincrono (force) un record che indica il completamento del checkpoint
- si riprende l'accettazione delle operazioni

checkpoint, versione C

- si scrive sul log in modo sincrono (force) un record che indica l'inizio del checkpoint, con gli identificatori delle transazioni in corso
- si aspetta che le transazioni attive raggiungano la conclusione, accettando comunque nuove transazioni
- quando tutte le transazioni attive all'inizio del checkpoint si concludono, si trasferiscono in memoria di massa (tramite force) tutte le pagine sporche relative a transazioni andate in commit
- si registra sul log in modo sincrono (force) un record che indica il completamento del checkpoint

Domanda 2 (20%) Si supponga di dover eseguire una interrogazione che calcola statistiche su una base di dati (ad esempio: trovare per ogni corso la media dei voti assegnati). Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi

1. l'interrogazione è eseguita in un tempo morto (cioè in assenza di aggiornamenti)
2. per tutti i corsi sono già presenti molti esami e l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); *non* sono accettabili risultati "approssimati"
3. per tutti i corsi sono già presenti molti esami e l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; *non* sono accettabili risultati "approssimati"
4. per tutti i corsi sono già presenti molti esami e l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; sono accettabili risultati "approssimati"
5. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, con alcuni corsi "nuovi," per i quali non vi è alcun esame memorizzato; sono accettabili risultati "approssimati"

Domanda 3 (25%) Si considerino due relazioni

- *Venditore(Matricola,Nome,Indirizzo,Budget)*
- *Vendita(Codice,Venditore,Importo)* (con vincolo di integrità referenziale fra *Venditore* e la relazione *Venditore*)

di dimensioni rispettivamente N_1 e $N_2 = k \times N_1$, con $k = 500$ e con ennuple rispettivamente di $l_1 = 100$ byte e $l_2 = 50$ byte, di lunghezza fissa. Si supponga che il sistema preveda blocchi di dimensione $B = 1000$ byte e che le relazioni abbiano una struttura hash su *Matricola* (la prima) e *Venditore* (la seconda). Valutare la convenienza dell'adozione della memorizzazione basata sul "co-clustering" (in cui un file contiene record di due o più relazioni e tali record sono allocati secondo i valori di opportuni campi dell'una e dell'altra relazione, nel nostro caso i campi *Matricola* e *Venditore*), rispetto al seguente insieme di operazioni (da considerare globalmente):

1. stampa del nome e di tutte le vendite di un venditore, data la matricola, con frequenza $f_A = 100$
2. stampa dell'elenco dei venditori, ordinato per *Nome*, con frequenza $f_B = 500$

Domanda 4 (15%) Spiegare perché il locking a due fasi stretto, con un commit globale, garantisce la serializzabilità in contesto distribuito mentre il il locking a due fasi locale non la garantisce.

Domanda 5 (10%) Discutere, in non più di una pagina, un concetto di ampio respiro che si ritiene di aver maturato attraverso il corso integrativo su "Data management in distributed multi-tier architectures." Si scelga liberamente il tema, tenendo presente che l'obiettivo fondamentale è la chiarezza e che si deve supporre di rivolgersi ad un lettore interessato ma non particolarmente competente sull'argomento (possibile scenario: lavorate in azienda e il vostro capo vi ha mandato a seguire il corso perché interessato all'argomento sulla base del titolo e ora vuole capire qual è la cosa più importante che avete imparato e pretende di capirla rapidamente anche lui).

Basi di dati, primo modulo — Tecnologia delle basi di dati

18 giugno 2003 — Compito B — Tempo a disposizione: due ore

Domanda 1 (30%) Considerare le seguenti tre versioni del checkpoint e commentare brevemente le differenze in termini di prestazioni e di procedure di ripristino dopo i guasti

checkpoint, versione A

- si sospende l'accettazione di nuove transazioni e si aspetta che le transazioni attive raggiungano la conclusione
- si trasferiscono in memoria di massa (tramite force) tutte le pagine sporche relative a transazioni andate in commit
- si scrive sul log in modo sincrono (force) un record che indica il completamento del checkpoint
- si riprende l'accettazione delle operazioni

checkpoint, versione B

- si sospende l'accettazione di richieste di ogni tipo (scrittura, inserimenti, ..., commit, abort)
- si trasferiscono in memoria di massa (tramite force) tutte le pagine sporche relative a transazioni andate in commit
- si scrive sul log in modo sincrono (force) un record che indica il completamento del checkpoint, con gli identificatori delle transazioni in corso
- si riprende l'accettazione delle operazioni

checkpoint, versione C

- si scrive sul log in modo sincrono (force) un record che indica l'inizio del checkpoint, con gli identificatori delle transazioni in corso
- si aspetta che le transazioni attive raggiungano la conclusione, accettando comunque nuove transazioni
- quando tutte le transazioni attive all'inizio del checkpoint si concludono, si trasferiscono in memoria di massa (tramite force) tutte le pagine sporche relative a transazioni andate in commit
- si registra sul log in modo sincrono (force) un record che indica il completamento del checkpoint

Domanda 2 (20%) Si supponga di dover eseguire una interrogazione che calcola statistiche su una base di dati (ad esempio: trovare per ogni corso la media dei voti assegnati). Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ` o `SERIALIZABLE`) si potrebbe scegliere in ciascuno dei seguenti casi

1. l'interrogazione è eseguita in un tempo morto (cioè in assenza di aggiornamenti)
2. per tutti i corsi sono già presenti molti esami e l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); sono accettabili risultati "approssimati"
3. per tutti i corsi sono già presenti molti esami e l'interrogazione è eseguita mentre vengono inseriti alcuni esami, comunque pochi per corso (rispetto a quelli già presenti); *non* sono accettabili risultati "approssimati"
4. per tutti i corsi sono già presenti molti esami e l'interrogazione è eseguita mentre vengono corretti (cioè modificati) i voti di alcuni esami, comunque pochi per corso (rispetto a quelli già presenti) e senza inserirne di nuovi; *non* sono accettabili risultati "approssimati"
5. l'interrogazione è eseguita mentre vengono inseriti alcuni esami, con alcuni corsi "nuovi," per i quali non vi è alcun esame memorizzato; sono accettabili risultati "approssimati"

Domanda 3 (25%) Si considerino due relazioni

- *Venditore(Matricola,Nome,Indirizzo,Budget)*
- *Vendita(Codice,Venditore,Importo)* (con vincolo di integrità referenziale fra *Venditore* e la relazione *Venditore*)

di dimensioni rispettivamente L_1 e $L_2 = k \times L_1$, con $k = 200$ e con ennuple rispettivamente di $l_1 = 100$ byte e $l_2 = 50$ byte, di lunghezza fissa. Si supponga che il sistema preveda blocchi di dimensione $B = 1000$ byte e che le relazioni abbiano una struttura hash su *Matricola* (la prima) e *Venditore* (la seconda). Valutare la convenienza dell'adozione della memorizzazione basata sul "co-clustering" (in cui un file contiene record di due o più relazioni e tali record sono allocati secondo i valori di opportuni campi dell'una e dell'altra relazione, nel nostro caso i campi *Matricola* e *Venditore*), rispetto al seguente insieme di operazioni (da considerare globalmente):

1. stampa dell'elenco dei venditori, ordinato per *Nome*, con frequenza $f_A = 100$
2. stampa del nome e di tutte le vendite di un venditore, data la matricola, con frequenza $f_B = 500$

Domanda 4 (15%) Spiegare perché, scrivendo un programma che accede a due basi di dati diverse utilizzando JDBC, non è possibile garantire il commit a due fasi. Indicare quali funzionalità aggiuntive sarebbero necessarie.

Domanda 5 (10%) Discutere, in non più di una pagina, un concetto di ampio respiro che si ritiene di aver maturato attraverso il corso integrativo su "Data management in distributed multi-tier architectures." Si scelga liberamente il tema, tenendo presente che l'obiettivo fondamentale è la chiarezza e che si deve supporre di rivolgersi ad un lettore interessato ma non particolarmente competente sull'argomento (possibile scenario: lavorate in azienda e il vostro capo vi ha mandato a seguire il corso perché interessato all'argomento sulla base del titolo e ora vuole capire qual è la cosa più importante che avete imparato e pretende di capirla rapidamente anche lui).