

02 Gennaio 2010

Domanda 1:

Il 2-phase-commit è un protocollo per il commit distribuito e permette a una transazione di prendere la decisione corretta di commit o abort su tutti i nodi che partecipano alla transazione.

Distinguiamo tra:

- RM - I server che partecipano alla decisione: gestori di risorse
- TM – Processo coordinatore: gestore della transazione

Si distinguono 2 fasi:

Fase 1:

TM

scrive il record prepare (contiene l'identità di tutti i processi RM) nel suo log e manda un msg prepare a tutti gli RM; imposta il timeout.

RM

Possono essere in 2 stati:

Affidabile (ha i lock su tutte le risorse che devono essere scritte):

Scriva il record ready (indica di voler partecipare al protocollo) e trasmette il msg ready.

Non Affidabile:

Invia msg non-ready e termina la partecipazione al protocollo

TM

Raccoglie i messaggi di risposta e se sono tutti ready iscrive nel log global commit altrimenti scrive global abort.

Fase 2:

TM

Trasmette la decisione (commit o abort) agli RM; imposta il timeout.

RM in stato ready

Scrivono nel log il record relativo alla decisione e mandano un ack al TM.

Eseguono in loco la decisione globale.

TM

Ricevuti tutti gli ack scrive il record complete (fine protocollo di commit a 2 fasi) nel log.

Domanda 2:

Nel locking a due fasi ogni transazione ha due fasi:

Crescente – acquisisce i lock

Decrescente – rilascia i lock

Una transazione dopo aver rilasciato un lock non può acquisirne altri (garantisce serializzabilità).

Uno Schedule 2PL base consiste in transazioni ben formate rispetto al locking, gestione dei lock basata sui conflitti e locking a due fasi.

Le varianti sono:

2PL stretto:

Una transazione può rilasciare i lock solo dopo il commit/abort (elimina possibilità di letture sporche).

2PL conservativo:

Una transazione deve acquisire tutti i lock prima di iniziare le operazioni (garantisce no deadlock).

Esempio di schedule non 2PL base:

S: $r_1(x) w_1(x) r_2(x) w_2(x) r_3(y) w_1(y)$

Domanda 3:

Il trigger, nelle basi di dati attive, è una procedura eseguita in maniera automatica in coincidenza di un determinato evento.

Se a seguito di un evento la condizione è soddisfatta esegue l'azione.

La creazione dei trigger fa parte del Data Definition Language.

I componenti sono:

Evento:

primitive SQL per la manipolazione dei dati (insert, delete, update).

Condizione:

predicato booleano espresso in SQL.

Azione:

sequenza di primitive SQL generiche o procedura.

Livelli di granularità:

Tupla:

l'azione avviene per ogni tupla coinvolta nell'evento.

Primitiva:

l'attivazione avviene una sola volta per ogni evento e si applica a tutte le tuple coinvolte nell'evento.

Modalità attivazione:

Immediata:

valutazione immediatamente dopo o prima dell'evento che lo ha attivato.

Differita:

valutazione avviene alla fine della transazione.

12 Gennaio 2010

Domanda 1:

Le proprietà ACIDE sono Atomicità, Consistenza, Isolamento e Durabilità.

Atomicità:

Una transazione è una unità atomica di lavoro.

Non può lasciare la base di dati in uno stato intermedio e un errore prima del commit causa l'UNDO del lavoro fatto fino a quel momento, mentre dopo il commit causa il REDO del lavoro.

Consistenza:

L'esecuzione di una transazione non deve violare i vincoli d'integrità definiti sulla base di dati. Il controllo dell'integrità può essere immediato (durante la transazione, l'operazione che causa la violazione è rifiutata) oppure differito (alla fine della transazione se vengono violati

dei vincoli, l'intera transazione è rifiutata).

Isolamento:

L'esecuzione di una transazione deve essere indipendente da quella di tutte le altre transazioni concorrenti.

Durabilità:

Gli effetti di una transazione che ha eseguito commit non devono essere persi.

La distribuzione dei dati l'isolamento e l'atomicità possono essere compromesse e sono richieste delle modifiche a:

- Ottimizzazione di interrogazioni
- Controllo di concorrenza (isolamento)
- Controllo di affidabilità (atomicità)

Controllo concorrenza:

Una transazione può eseguire più sotto-transazioni a diversi nodi.

Atomicità:

Può essere compromessa da guasti o malfunzionamenti.

Domanda 2:

- 1 Permette di assegnare timestamp che riflettono la precedenza fra eventi in un sistema distribuito. I timestamp hanno due gruppi di cifre X.Y (X identifica l'evento che accade al nodo e Y identifica un nodo). Ogni volta che i nodi si scambiano un messaggio, i timestamp vengono sincronizzati: l'evento ricevente deve avere un timestamp maggiore o uguale dell'evento mittente (può richiedere l'incremento del contatore locale al nodo ricevente).
- 2a Prevenzione deadlock:
 - Preemptive (interrompenti): uccidono la transazione che possiede la risorsa.
 - Non-preemptive (non interrompenti): uccidono la transazione che richiede la risorsa.
- 2b Le transazioni uccise devono ripartire con lo stesso timestamp (altrimenti rischierebbero di essere sempre uccise (starvation)).
- 3 La frammentazione dei dati applica delle operazioni algebriche su una relazione R per dividerla in frammenti.
 - Frammentazione orizzontale:
 - Ogni frammento ha come tuple un sottoinsieme delle tuple della relazione e può essere interpretato come il risultato di una selezione sulla relazione.
 - (i frammenti sono disgiunti quindi non hanno tuple in comune)
 - Frammentazione verticale:
 - Ogni frammento ha come schema un sottoinsieme degli attributi della relazione e può essere interpretato come il risultato di una proiezione sulla relazione.
 - (i frammenti includono la chiave primaria per garantire la ricostruibilità)
 - Proprietà di correttezza:
 - Completezza:
 - ogni dato della relazione deve essere presente in un qualche suo frammento.
 - Ricostituibilità:
 - deve essere interamente ricostruibile a partire dai suoi frammenti.

11 Settembre 2009

Domanda 1:

I guasti a un protocollo two-phase commit posso essere:

- 1 - caduta di un RM
- 2 - caduta del TM
- 3 - perdita di messaggi
- 4- partizionamento della rete

Caduta di un RM:

RM esegue la ripresa a caldo e se l'ultimo record era ready deve chiedere al TM, per ogni transazione in dubbio, la decisione finale.

Caduta del TM:

TM esegue la ripresa a caldo e guarda l'ultimo record di log:

Prepare: o scrive un global abort nel log ed esegue la seconda parte del protocollo oppure ripete la prima fase.

Global commit/abort: deve ripetere la seconda fase.

Complete: non ha effetti sulla transazione.

Perdita di messaggi:

Quando viene perso il messaggio di prepare o di ready scade il timeout e viene presa la decisione globale di abortire.

Quando viene perso il messaggio commit/abort o di un ack scade il timeout e viene ripetuta la seconda fase.

Partizione della rete:

La transazione avrà successo solo se il TM e gli RM appartengono alla stessa partizione.

Domanda 2:

1 Ogni pagina sequenziale ha (assumendo un blocco per pagina):

- 1) Una parte iniziale (block header) e una parte finale (block trailer) contenenti l'informazione di controllo usata dal file system.
- 2) Una parte iniziale (page header) e una parte finale (page trailer) contenenti l'informazione del metodo di accesso.
- 3) Un dizionario di pagina che contiene i puntatori a ogni dato elementare contenuto nella pagina.
- 4) Una parte utile che contiene i dati.
- 5) Un bit di parità per verificare l'integrità dell'informazione contenuta nella pagina.

Le primitive sono:

- 1) Inserimento e aggiornamento di una tupla.
- 2) Cancellazione di una tupla.
- 3) Accesso ad una specifica tupla.
- 4) Accesso ad un campo di una specifica tupla.

2 Il modello multidimensionale permette la rappresentazione dei dati ad alto livello e favorisce l'analisi.

Concetti rilevanti:

- 1) Fatto: concetto sul quale concentrare l'analisi.
- 2) Misura: proprietà atomica di un fatto da analizzare.
- 3) Dimensione: descrivere una prospettiva lungo la quale effettuare l'analisi.

Esempio:

Catena di negozi

Fatto: vendita

Misura: quantità venduta, incasso

Dimensione: articolo, tempo, luogo

3 I livelli di isolamento riguardano le letture perché la scrittura a ogni livello richiedono 2PL stretto.

I livelli sono:

- 1) Read uncommitted: Non chiede lock per leggere, non rispetta i lock esclusivi delle altre transazioni e può presentare tutte le anomalie delle transazioni concorrenti.
- 2) Read committed: Chiede lock condivisi per le letture, non legge dati non committed, evita letture sporche ma non garantisce serializzabilità.
- 3) Repeatable Read: Chiede 2PL stretto anche per la lettura, lock a livello di tupla, evita tutte le anomalie ad eccezione dell'inserimento fantasma.
- 4) Serializable: Chiede 2PL stretto anche per la lettura e utilizza lock di predicato, evita tutte le anomalie.

10 Luglio 2009

Domanda 1:

Le proprietà di Atomicità e Isolamento sono intaccate dalla distribuzione dei dati.

Esempio Atomicità:

CC(Num, Nome, Saldo) Frammentato in CC1 (con num ≤ 1000) e CC2 (con num > 1000)

Assumiamo trasparenza di allocazione.

Se dobbiamo trasferire 100 Euro dal conto 354 al 1487 dobbiamo riuscire a garantire l'atomicità quindi o entrambe le modifiche sono eseguite o non lo è nessuna.

begin transaction

```
update CC1
set saldo = saldo - 100
where CCNum = '354';
```

```
update CC2
set saldo = saldo + 100
where CCNum = '1487';
```

```
commit;
end transaction
```

Esempio Isolamento:

In un sistema distribuito una transazione T_i può eseguire più sotto-transazioni a diversi nodi:

T₁ = R₁₁(x) W₁₁(x) R₁₂(y) W₁₂(y)

T₂ = R₂₂(x) W₂₂(y) R₂₁(x) W₂₁(x)

Localmente risultano serializzabili ma globalmente no infatti T₁ sul nodo 1 precede T₂ mentre T₂ sul nodo 2 precede T₁.

Per risolvere il problema sono richieste le seguenti modifiche: ottimizzazione di interrogazioni, controllo di concorrenza e controllo di affidabilità.

Domanda 2:

In molti sistemi reali i lock possono essere chiesti su oggetti a granularità differente. Su una tupla possiamo chiedere:

- 1) XL (exclusive lock): corrisponde al write lock
- 2) SL (shared lock): corrisponde al read lock
- 3) ISL (intentional shared lock): intenzione di bloccare in modo condiviso uno dei nodi che discende dal nodo corrente.
- 4) IXL (intentional exclusive lock): intenzione di bloccare in modo esclusivo uno dei nodi che discende dal nodo corrente.
- 5) SIXL (shared intentional-exclusive lock): chiede lock condiviso sul nodo corrente e esprime l'intenzione di bloccare in modo esclusivo uno dei nodi che discende dal nodo corrente.

Protocollo:

- 1) I lock si richiedono dalla radice e scendendo lungo l'albero.
- 2) I lock si rilasciano dal nodo locked e salendo lungo l'albero.
- 3) Una transazione può richiedere un lock SL o ISL su un nodo, solo se ha un lock ISL o IXL sul genitore.
- 4) Una transazione può richiedere un lock IXL, XL, o SIXL su un nodo, solo se ha già un lock SIXL o IXL sul genitore.

Domanda 3:

1 Supporto: Probabilità che siano presenti in una transazione entrambi gli elementi di una regola.

Confidenza: Probabilità che sia presente in una transazione la conseguenza di una regola, essendo presente la premessa.

Esempio: pannolini → birra

Supporto = 0.02: il 2% tra tutte le transazioni contiene sia pannolini sia birra.

Confidenza = 0.30: il 30% delle transazioni che contiene pannolini contiene anche la birra.

2 Il metodo di lamport si utilizza per garantire la serializzabilità globale nelle base di dati distribuite. Permette di assegnare timestamp che riflettono la precedenza fra eventi in un sistema distribuito.

Timestamp: X.Y (X = contatore locale , Y = identificano un nodo)

ogni volta che due nodi si scambiano un messaggio i timestamp vengono sincronizzati e l'evento ricevente deve avere un timestamp maggiore o uguale all'evento mittente.

3 Checkpoint: Operazione di sistema svolta dal gestore dell'affidabilità con il coordinamento del buffer manager. Registra nel log le transazioni attive e aggiorna la memoria secondaria rispetto tutte le transazioni complete (è eseguito periodicamente).

Dump: Copia completa della base di dati memorizzata su memoria stabile (backup). Viene creata quando il sistema non è operativo. Alla conclusione del dump viene scritto nel log il record di dump.

Memoria stabile: Memoria resistente ai guasti. E' un astrazione poiché nessuna memoria può

avere probabilità di guasto 0. Ma è possibile renderla prossima allo 0 tramite replicazione e protocolli di scrittura robusti.

4: VSR ma non CSR:

Schedule: $r1(x), r2(y), w1(x), w2(y), w1(y)$

4 Novembre 2009

Domanda 1:

Il modello multidimensionale permette la rappresentazione dei dati ad alto livello favorendone l'analisi.

I concetti rilevanti sono:

- 1) Fatto: concetto sul quale centrare l'analisi
- 2) Misura: proprietà atomica di un fatto da analizzare
- 3) Dimensione: descrive una prospettiva lungo la quale effettuare l'analisi.

Le operazioni sono:

- 1) Slice-and-dice (selezione e proiezione): seleziona un sottoinsieme delle celle di un cubo.
- 2) Roll-up (aggrega i dati): applica una funzione aggregativa sui dati aggregati di un cubo.
- 3) Drill-down (disaggrega i dati): è l'operazione inversa del roll-up. Disaggrega il cubo lungo una o più dimensioni.

La rappresentazione tramite schema a stella ha come componenti:

- 1) Tabella dei fatti: tabella principale che memorizza i fatti del data mart.
- 2) Tabella dimensione: relazioni ausiliarie che memorizzano i dati relativi alle dimensioni dell'analisi.
- 3) Vincoli di integrità referenziale: ognuno collega un attributo della tabella fatti a una tabella dimensione.

La rappresentazione tramite schema a fiocco di neve:

- 1) Risulta da una normalizzazione di uno schema a stella.
- 2) Permette di evitare ridondanze eccessive nelle dimensioni
- 3) Dalla tabella dei fatti si raggiungono tutte le tabelle delle dimensioni

Domanda 2:

- 1 Write Ahead Log: La parte BS dei record di log deve essere scritta nel log prima di effettuare la corrispondente operazione sulla base di dati. Consente undo delle scritture effettuate da transazioni che non hanno fatto commit.
Commit-Precedenza: La parte AS dei record di log deve essere scritta nel log prima di effettuare il commit. Consente redo delle scritture già decise da transazioni che hanno fatto commit, ma le cui pagine modificate non sono ancora state scritte dal buffer manager su memoria secondaria.
- 2 Il deadlock nelle basi di dati distribuite sono circoli di attesa tra due nodi o più nodi e due possibili soluzioni sono il time-out e la rilevazione e risoluzione. La rilevazione può essere fatta con un protocollo asincrono e distribuito. Attivata periodicamente ai vari DBMS del sistema, ogni DBMS:
 - 1) integra nuove sequenze di attesa con le condizioni di attesa locali
 - 2) analizza il grafo risultante per rilevare deadlock
 - 3) comunica le sequenze di attesa ad altri DBMS

4 XML Schema elementi semplici:

- 1) non possono contenere altri elementi o attributi
- 2) possono essere solo di tipo standard o tipi di dati derivati da questi.
- 3) Possono avere valori di default
- 4) possono avere valori fissi
- 5) possono avere associate delle restrizioni

Esempio:

```
<xs:element name="nome" type="tipo" />
```

XML Schema: elementi complessi

- 1) possono contenere attributi o altri elementi
- 2) possono utilizzare indicatori

Esempio:

```
<xs:element name="persona">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Cognome" type="xs:string" />
      <xs:element name="Nome" type="xs:string"
        maxOccurs="4" minOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XML Schema: indicatori

- 1) Ordinamento:
 - Any: qualunque elemento, in qualunque ordine
 - All: tutti gli elementi, in qualunque ordine
 - Choice: uno e un solo elemento
 - Sequence: tutti gli elementi, nell'ordine specificato
- 2) Cardinalità (numero di occorrenze): MaxOccurs e MinOccurs.
- 3) Raggruppamento: Group name e Group reference.

9 Gennaio 2009

Domanda 1:

L'organizzazione fisica regola la disposizione delle tuple all'interno del file:

- 1) Sequenziale: tuple disposte in modo sequenziale
- 2) Accesso calcolato: posizione delle tuple determinata da un algoritmo di calcolo.
- 3) A indici: organizzazione ad albero

Strutture sequenziali tipologie:

- 1) Seriale: la sequenza delle tuple è indotta dal loro ordine di inserimento.
- 2) Ad array: le tuple sono disposte in un array e la loro posizione dipende dal valore di uno più campi indice.
- 3) Sequenziale ordinata: la sequenza delle tuple dipende dal valore, in ciascuna tupla, di un campo di ordinamento (campo chiave).

Scritture seriale:

Vantaggi: ottimale per le operazioni di scrittura e le letture sequenziali.

Svantaggi: le cancellazioni possono lasciare spazio non utilizzato, le modifiche creano

aumento delle dimensioni della tupla non possono essere gestite in loco e l'accesso ai singoli dati richiede la scansione dell'intero file.

Scrittura ad array:

Vantaggi: accesso diretto (sulla base della chiave)

Svantaggi: possibile solo quando le tuple hanno lunghezza fissa, le cancellazioni lasciano celle vuote e si può applicare solo quando la chiave assume valori consecutivi.

Sequenziale ordinata:

Vantaggi: efficienti operazioni basate sul valore della chiave.

Svantaggi: inserimenti e modifiche richiedono la riorganizzazione delle tuple già presenti.

Domanda 3:

Proprietà delle regole attive:

- 1) Terminazione: Per qualunque stato iniziale e qualunque sequenza di modifiche, le regole producono uno stato finale (no cicli infiniti).
- 2) Confluenza: Le regole producono uno stato finale, producono un unico stato finale, indipendente dall'ordine in cui trigger vengono eseguiti.
- 3) Determinismo: Le regole terminano e producono un unico stato finale, indipendente dall'ordine in cui le regole vengono eseguite, producono la stessa sequenza di azioni visibili.

11 Giugno 2009

Domanda 3

Two-Phase Locking: Una transazione dopo aver rilasciato un lock, non può acquisirne altri. Ogni transazione ha due fasi una crescente (acquisita i lock) e una decrescente (rilascia i lock). Questa tecnica garantisce serializzabilità.

Two-Phase commit: Permettono a una transazione di raggiungere la decisione corretta di commit o abort su tutti i nodi che partecipano alla transazione.