Ottimizzazione delle interrogazioni (parte I)

Angelo Montanari

Dipartimento di Matematica e Informatica
Università di Udine

Introduzione

Il modulo **ottimizzatore** decide le strategie di accesso ai dati in fase di esecuzione delle interrogazioni

L'ottimizzatore è un componente classico (e fondamentale) di un DBMS

L'ottimizzatore riceve in input un'interrogazione SQL, trasmessa attraverso uno qualunque dei meccanismi dell'interfaccia applicativa disponibili (cursori, procedure, ..)

Attività preliminari

Il modulo, facendo uso del dizionario dei dati, esegue un'**analisi della struttura** lessicale, sintattica e semantica dell'interrogazione

Se rileva degli errori lessicali (errori tipografici), sintattici (interrogazione malformata) o semantici (il valore di un attributo viene confrontato con una costante che non appartiene al dominio dell'attributo), li segnala all'utente per la correzione dell'interrogazione

Dal dizionario dei dati, l'ottimizzatore legge anche alcune **informazioni di natura statistica** che riguardano la dimensione delle tabelle

Una volta accettata, l'interrogazione viene tradotta in una **forma interna di tipo algebrico**

Ottimizzazione algebrica

Viene quindi eseguita la vera e propria attività di ottimizzazione che si articola in due fasi successive:

1. **Ottimizzazione algebrica**: esegue un insieme di trasformazioni algebriche (ad esempio, lo spostamento delle operazioni di selezione e proiezione verso le foglie dell'albero che rappresenta la struttura dell'interrogazione), che consentono di eseguire l'interrogazione in modo più efficiente senza alterarne la semantica

   È un'ottimizzazione di tipo logico che avviene in modo del tutto indipendente dal modello dei costi prescelto

Ottimizzazione basata sui costi

2. **Ottimizzazione basata sui costi**: esegue una ottimizzazione che dipende sia dalla **tipologia dei metodi di accesso ai dati** supportati dal livello (fisico) sottostante sia dal **modello dei costi** prescelto

   Questa fase dipende dalle caratteristiche peculiari del sistema

Ultimato il processo di ottimizzazione, si procede alla **generazione del codice**, che utilizza i metodi di accesso ai dati messi a disposizione dal sistema. I **programmi di accesso** generati sono in un formato interno, o formato oggetto, e utilizzano le strutture dati messe a disposizione dal sistema, in particolare gli indici

Compilazione di un'interrogazione

Riassumendo, il processo di ottimizzazione di un'interrogazione riceve in input interrogazioni in SQL e restituisce in output programmi di accesso, con associato il relativo **insieme di dipendenze** (dipendenze del codice dalle particolari versioni delle tabelle e degli indici della base di dati, letti nel dizionario dei dati)

Nell'ordine vengono eseguite (i) l'analisi lessicale, sintattica e semantica, che utilizza informazioni presenti nel dizionario dei dati, (ii) l'ottimizzazione algebrica, (iii) l'ottimizzazione basata sui costi di esecuzione, che utilizza informazioni relative ai **profili** delle tabelle e delle operazioni e (iv) la generazione del codice

Approccio compile-and-store vs. compile-and-go

A differenza degli altri moduli del DBMS, l'ottimizzatore agisce al tempo di compilazione

Approccio compile-and-store: se l'interrogazione viene compilata una volta ed eseguita molte volte, il codice viene prodotto e memorizzato nella base di dati insieme alle dipendenze

La memorizzazione delle dipendenze consente di rilevare cambiamenti della base di dati significativi per l'interrogazione (ad esempio, l'aggiunta di un nuovo indice), che suggeriscono/ impongono la ricompilazione dell'interrogazione

Approccio compile-and-go: se l'interrogazione viene compilata ed eseguita una sola volta, il codice prodotto non viene memorizzato

Le due fasi del processo di ottimizzazione

1. Nella **prima fase** viene eseguita un'ottimizzazione algebrica che produce una descrizione ottimizzata dell'interrogazione, in cui sono state effettuate tutte le possibili trasformazioni algebriche (cf. algebra relazionale)

   Il risultato tale fase rappresenta ogni interrogazione SQL in strutture ad albero, in cui i nodi foglia rappresentano tabelle e i cui nodi intermedi rappresentano operazioni dell'algebra relazionale

2. Nella **seconda fase** viene eseguita un'ottimizzazione basata sui costi. Essa dipende fortemente dalle specifiche strutture di memorizzazione utilizzate e dal modello dei costi prescelto. È, comunque, possibile fornire una descrizione generale qualitativa di tale fase

Profili delle relazioni

Ogni DBMS commerciale possiede informazioni quantitative relative alle caratteristiche delle tabelle organizzate in strutture dati dette **profili delle relazioni**, che vengono memorizzate nel dizionario dei dati

I profili contengono (parte del) le seguenti informazioni:

- la cardinalità $CARD(T_i)$ (numero di tuple) di ogni tabella T_i
- la dimensione in byte $SIZE(T_i)$ di ogni tupla di T_i
- la dimensione in byte $SIZE(A_j)$ di ciascun attributo A_j
- il numero di valori distinti $VAL(A_j)$ di ogni attributo A_j
- i valori minimo $MIN(A_j)$ e massimo $MAX(A_j)$ di ogni attributo A_j

Sul calcolo dei profili

Come (e quando) vengono calcolati i profili?

I profili vengono calcolati sulla base dei dati effettivamente memorizzati nelle tabelle, richiamando opportune primitive di sistema (ad esempio, la primitiva *update statistics*). Tali primitive vengono richiamate periodicamente dall'amministratore della base di dati

Risulta troppo costoso tenere aggiornati i profili durante la normale esecuzione delle transazioni

È sufficiente che i profili contengano valori approssimati (anche perché i modelli statistici ad essi applicati sono anch'essi approssimati)

I profili delle operazioni

Non basta calcolare i profili delle relazioni (tabelle) memorizzate nella base di dati ...

L'ottimizzazione dipendente dai costi richiede di formulare delle ipotesi sulle **dimensioni dei risultati intermedi** prodotti dalla valutazione di operazioni algebriche con un approccio di tipo statistico

Sono state proposte delle regole per il calcolo dei profili relativi alle operazioni algebriche di base (selezione, proiezione e join)

I profili delle operazioni: selezione - 1

Formule dei profili relativi all'operazione di selezione. Il profilo di una tabella T prodotta da una selezione $T = \sigma_{A_i=v}(R)$ è ottenuto mediante le seguenti formule:

1. $CARD(T) = CARD(R) \cdot 1/VAL(A_i)$;
2. $SIZE(T) = SIZE(R)$;
3. $VAL(A_i) = 1$;
4. $VAL(A_j) = col(CARD(R), VAL(A_j), CARD(T))$, con $j \neq i$;
5. $MAX(A_i) = MIN(A_i) = v$;
6. $MAX(A_j)$ e $MIN(A_j)$, con $j \neq i$ rimangono invariati.

I profili delle operazioni: selezione - 2

La funzione $col(n, m, k)$ usata nella formula che definisce il nuovo valore di $VAL(A_j)$, con $j \neq i$, calcola il numero di colori (valori) distinti di k oggetti estratti da un insieme di n oggetti di m colori (valori) distinti distribuiti in modo omogeneo.

Essa ammette la seguente approssimazione:

- $col(n, m, k) = k$, se $k \leq m/2$;
- $col(n, m, k) = \lceil (k + m)/3 \rceil$, se $m/2 \leq k \leq 2 \cdot m$;
- $col(n, m, k) = m$, se $k \geq 2 \cdot m$.

I profili delle operazioni: proiezione

Formule dei profili relativi all'operazione di proiezione. Il profilo di una tabella T prodotta da una proiezione $T = \Pi_L(R)$, dove L è la sequenza di attributi $A_1, \ldots, A_n$, è ottenuto mediante le seguenti formule:

1. $CARD(T) =$
   $MIN(CARD(R), VAL(A_1) \cdot VAL(A_2) \cdot \ldots VAL(A_n))$;
2. $SIZE(T) = SIZE(A_1) + SIZE(A_2) + \ldots + SIZE(A_n)$;
3. $VAL(A_i)$, $MAX(A_i)$ e $MIN(A_i)$ rimangono invariati.

I profili delle operazioni: join

Formule dei profili relativi all'operazione di join. Il profilo di una tabella T prodotta da un equi-join $T = R \bowtie_{A=B} S$, assumendo che A e B abbiano il medesimo dominio e che $VAL(A) = VAL(B)$, è ottenuto mediante le seguenti formule:

1. $CARD(T) = CARD(R) \cdot CARD(S) \cdot 1/VAL(A)$;
2. $SIZE(T) = SIZE(R) + SIZE(S)$;
3. $VAL(A_i)$, $MAX(A_i)$ e $MIN(A_i)$ rimangono invariati.

Nota bene: qualora si esegua un natural join sull'(unico) attributo comune A, anziché un equi-join, la seconda formula va riscritta come segue:

$SIZE(T) = SIZE(R) + SIZE(S) - SIZE(A)$;

Limiti

Il calcolo dei profili delle operazioni di selezione, proiezione e join mostra i limiti di questo tipo di analisi statistica.

Tutte le formule assumono una **distribuzione uniforme** dei dati nelle tabelle e un'**assenza di correlazioni** fra le varie condizioni presenti in un'interrogazione.

Inoltre, spesso le formule assegnano al risultato di un'operazione valori identici a quelli dei loro operandi in quanto non si dispone di informazioni più precise sugli effetti dell'operazione (ad esempio, valori massimi e minimi di un dato attributo).

Ciò nonostante, tale analisi definisce l'**ordine di grandezza** delle dimensioni dei risultati intermedi (ad esempio, in termini di numero di pagine occupate), sufficiente dal punto di vista dell'ottimizzazione.