

FORMULE DI BASI DI DATI

14 luglio 2021

FORMULE SUI B-TREE

DEFINIZIONE DI B-TREE :

Un B-TREE è un albero bilanciato, organizzato a nodi, ogni nodo corrisponde ad un blocco dati di uno storage device

IMPORTANTE :

$T(g,h)$ è un albero bilanciato di ordine g e altezza h . Le cardinalità/configurazioni possibili sono rappresentate nella tabella sottostante. Ricordiamo che : $|sk|$ = numero di chiavi selezionate per nodo(nodo != radice).

Numero Massimo e minimo di chiavi per ogni nodo o radice:

CHIAVI	MIN	MAX
Radice	1	$2g$
Nodo	g	$2g$

Numero Massimo e minimo di figli per ogni nodo o radice:

FIGLI	MIN	MAX
Radice	0	$2g+1$
Nodo	$ sk + 1 $	$ sk + 1 $

FORMULE :

NUMERO MINIMO DI NODI(IP_{min})

$$IP_{min} = 1 + 2 \cdot \sum_{i=0}^{h-2} (g + 1)^i$$

NUMERO MASSIMO DI NODI(IP_{max})

$$IP_{max} = \sum_{i=0}^{h-1} (2g + 1)^i$$

ALTEZZA DI UN B-TREE

$NK = \text{Numero di chiavi del B-TREE}$

$$NK_{min} = 1 + g(IP_{min} - 1) = 2(g + 1)^{h-1} - 1$$

$$NK_{max} = 2g(IP_{max}) = (2g + 1)^h - 1$$

$$h_{min} = \log_{2g+1}(NK + 1)$$

$$h_{max} = 1 + \log_{g+1}\left(\frac{NK + 1}{2}\right)$$

$$h_{min} \leq h \leq h_{max}$$

COSTO INSERIMENTO/ELIMINAZIONE DI UN NODO DA UN B-TREE(g)

CASO :	MIGLIORE	PEGGIORE	MEDIA
Inserimento	$h + 1$	$3h + 1$	$h + 1 + 2/g$
Eliminazione	$h + 1$	$3h$	$5h + 5 + 3/g$

STIMA VARIABILITA' DELL'ORDINE DI UN B-TREE

$k = \text{chiave}$

$p = RID = \text{Record Identifier, ovvero puntatore a record}$

$q = PID = \text{Puntatore al nodo figlio}$

$D = \text{page size}$

$$2g * len(k) + 2g * len(p) + (2g + 1) * len(q) \leq D$$

$$g = \frac{D - len(q)}{2(len(k) + len(p) + len(q))}$$

FORMULE SUI B+-TREE

DEFINIZIONE DI B+-TREE :

Un B+-TREE è un B-TREE in cui i record pointer sono memorizzati solo nei nodi foglia dell'albero. La struttura dei nodi foglia differisce quindi da quella dei nodi interni. Ha prestazioni migliori del B-TREE sulla ricerca sequenziale, peggiori nella ricerca per singolo valore.

ORDINE

k = chiave

q = PID = Puntatore al nodo figlio

D = page size

$$2g * len(k) + (2g + 1) * len(q) \leq D$$

$$g = \frac{D - len(q)}{2(len(k) + len(q))}$$

NUMERO DI FOGLIE

NR = Numero di Record

NL = Numero di foglie

u = % di utilizzo di un singolo nodo (in media è il 69%)

d = dimensione dei nodi

$$NL = \frac{NR \cdot (len(k) \cdot len(q))}{d \cdot u}$$

ORDINE DELLE FOGLIE

D = page size

$$2g_{leaf} * len(k) + 2g_{leaf} * len(p) + (2g_{leaf} + 1) * len(q) \leq D$$

$$g_{leaf} = \frac{D - len(q)}{2 \cdot (len(k) + len(p))}$$

Con questo risultato si è in grado di avere una stima più accurata del numero di foglie di un B+-TREE

NUMERO DI FOGLIE (stima più accurata sfruttando g_{leaf})

$$NL = \frac{NR}{2g_{leaf} \cdot u}$$

ALTEZZA

Costruendo un albero in cui ciascun nodo ha il numero massimo di figli, si minimizza l'altezza del B+-TREE

$$(2g + 1)^{h-1} \geq NL$$

$$h_{min} = 1 + \log_{2g+1} NL$$

Similmente, sfruttando lo stesso ragionamento, costruendo un albero con il numero minimo possibile di figli per nodo, si massimizza l'altezza

$$2(g + 1)^{h-2} \leq NL$$

$$h_{max} = 1 + \log_{2g+1} NL$$

$$h_{min} \leq h \leq h_{max}$$

RICERCA DI VALORI

Supponendo di avere un B-TREE con NK chiavi in NL foglie, si effettua una ricerca sequenziale di k chiavi nell'intervallo di valori compresi fra $[k_{low}, k_{high}]$

EK = Numero di chiavi all'interno dell'intervallo $[k_{low}, k_{high}]$

EL = expected leafs, ovvero è una stima del numero di foglie alla quale si deve accedere durante la ricerca

$$EL = \frac{EK \cdot NK}{NL} \text{ (proporzione } \rightarrow NK : NL = EK : EL)$$

$$\text{COSTO(Ricerca)} = 1 - h + EL$$

NUMERO DI FOGLIE DI UN SECONDARY B+-TREE

$$NL = \frac{NK \cdot \text{len}(k) + NR \cdot \text{len}(p)}{D * u}$$

ALTEZZA DI UN SECONDARY B+-TREE

$$h_{min} = 1 + \log_{2g+1} \min(NL, NK)$$

$$h_{max} = 2 + \log_{g+1} \frac{\min(NL, NK)}{2}$$

$$h_{min} \leq h \leq h_{max}$$

MODELLO DI CARDENAS

Il modello di Cardenas è utile per stimare il numero medio di pagine NP che contengono almeno uno degli ER (expected records) presi in considerazione.

Considerando i seguenti eventi :

A = "Una pagina contiene 1 degli ER record"

\overline{A} = "Una pagina **non** contiene 1 degli ER record"

B = "Una pagina **non** contiene **nessuno** degli ER record "

\overline{B} = "Una pagina contiene almeno 1 degli ER record"

Definiamo :

$$P(A) = \frac{1}{NP} \quad P(\overline{A}) = 1 - \frac{1}{NP}$$

$$P(B) = P(\overline{A})^{ER} \quad P(\overline{B}) = 1 - P(B)$$

FORMULA DI CARDENAS :

$$\phi(ER, NP) = NP \cdot P(\overline{B})$$

$$= NP * (1 - (1 - \frac{1}{NP})^{ER}) \leq MIN(ER, NP)$$

QUERY PLAN

Gestione del piano di accesso alle query, l'obiettivo è quello di scegliere il metodo più veloce ed efficiente.

Linear counting

Algoritmo che permette di stimare il numero di NK chiavi distinte di un attributo A, con $A \in r$.

r = una qualsiasi estensione di una relazione $R(T)$.

DATI :

BM = mappa chiave - valore

H = hash function

$t_i.A$ = valore dell'elemento $\in A$ di una determinata tupla.

B = n. totale di bucket della bitMap

Z = n. di bucket della bitMap che hanno value = 0

Nota :

Ogni elemento $t_i.A$ sarà presente all'interno della bit-map BM seguendo la seguente logica:

if ($t_i.A == \text{NULL}$); then $\text{BM}[\text{H}(t_i.A)] = 0$;
else then $\text{BM}[\text{H}(t_i.A)] = 1$;

FORMULA RISOLUTIVA :

$$NK^e = -B * \ln\left(\frac{Z}{B}\right)$$

PROCEDIMENTO

Considerando l'evento $A = \text{"un bucket contiene ALMENO un valore"}$.

dal modello di cardenas si avrà :

$$P(A) = (1 - (1 - (\frac{1}{B})^{NK}))$$

$$P(\bar{A}) = (1 - (\frac{1}{B})^{NK})$$

il quale ci permetterà di formulare :

$$B - Z = B \cdot P(A) = \textit{stima del n. di bucket pieni}$$

$$Z = B \cdot P(\bar{A}) = \textit{stima del n. di bucket vuoti}$$

Successivamente è importante eseguire la seguente approssimazione, applicabile solo nel caso in cui si lavori con valori molto grandi :

$$P(\bar{A}) \simeq e^{-\frac{NK}{B}} = P(\bar{A}_{\sim})$$

di conseguenza Z diventerà :

$$Z = B \cdot P(\bar{A}_{\sim}) = B \cdot e^{-\frac{NK}{B}}$$

A questo punto, è possibile estrarre NK , il valore cercato.

$$NK^e = -B * \ln(\frac{Z}{B})$$

l'apice e aggiunto ad NK sta a significare estimated, ovvero il valore stimato, non necessariamente esatto, è stato aggiunto nel risultato per non confonderlo con la costante di Eulero.

Selettività dei predicati

f_p = fattore di selettività; $0 \leq f_p \leq 1$

$ER = f_p \cdot NR = \text{expectedRecords}$

da cui :

$f_p = \frac{EK}{ER} = \text{valoriSelezionati/valoriTotali}$

Casi notevoli

Predicato	Formula
=	$f_p = \frac{1}{NK}$
IN	$f_p = \frac{\text{card(Set)}}{NK}$
<	$f_p = \frac{v - \min(R.A)}{\max(R.A) - \min(R.A)} \cdot \frac{NK-1}{NK}$
BETWEEN	$f_p = \frac{v2-v1}{\max(R.A) - \min(R.A)} \cdot \frac{NK-1}{NK} + \frac{1}{NK}$

Predicati composti

$p = (p_1 \text{ AND } p_2) \hookrightarrow f_p = f_{p1} \cdot f_{p2}$

$p = (p_1 \text{ OR } p_2) \hookrightarrow f_p = f_{p1} + f_{p2} - f_{p1} \cdot f_{p2}$

$p = \overline{p_1} \hookrightarrow f_p = 1 - f_{p1}$

Costo di un query plan

full table scan

$$C(SeqR) = C_{I/O}(SeqR) = NP \cdot \alpha \cdot NR$$

Generale : IX(A) index scan

$$EN + EL + EP + \alpha \cdot ER$$

Scan con indice clustered

Su singolo valore o range:

$$EN = h - 1$$

$$EL = f_{p(A)} \cdot NL$$

$$EP = f_{p(A)} \cdot NP$$

Sul set :

$$EN = (h - 1) \cdot EK$$

$$EL = EK \cdot \frac{NL}{\overline{NK}}$$

$$EP = EK \cdot \frac{NP}{\overline{NK}}$$

Scan con indice unclustered

Su singolo valore o range :

$$EN = h - 1$$

$$EL = f_p \cdot NL$$

$$EP = EK \cdot \phi\left(\frac{NR}{\overline{NK}}, NP\right)$$

Su set :

$$EN = (h - 1) \cdot EK$$

$$EL = EK \cdot \frac{NL}{\overline{NK}}$$

$$EP = EK \cdot \phi\left(\frac{NR}{\overline{NK}}, NP\right)$$

Proiezione

SELECT DISTINCT <select list> FROM R
Y = (R.A₁, R.A₂, ..., R.A_N) (attributi citati nella select list)

Modalità di calcolo di ER(Expected records)

$ER_{\pi_{Y(R)}} = NR_R$ (i vari $NK_{R.A}$ non sono noti, si assume il caso peggiore)

$ER_{\pi_{Y(R)}} = NK_{R.A}$ (se la proiezione riguarda un solo attributo)

$ER_{\pi_{Y(R)}} = \min(NR_r, \prod_i NK_{R.A_i})$ (caso peggiore : si assume indipendenza tra gli attributi che fanno parte di Y (non superchiave))

fattori di selettività della proiezione

$f_{\pi_{Y(R)}} = 1$ (se la proiezione contiene una chiave candidata)

$f_{\pi_{Y(R)}} = \frac{NK_a}{NR_R}$ (se la proiezione riguarda un solo attributo A)

$f_{\pi_{Y(R)}} = \frac{\min(NR_r, \prod_i NK_{R.A_i})}{NR_R}$ (si assume indipendenza fra gli attributi)

cardinalità di una proiezione multi attributo

$\phi(NR, NK_{R.A} \cdot NK_{R.B})$

Proiezione basata su sorting

1. Full scan di R e creazione della table T, che conterrà gli elementi selezionati dalla `select(ResultSet)`.

$$C(\text{seq R and build T}) = NP_R + \alpha \cdot NR_R + EP_T$$

2. Sia EP_T il numero stimato di pagine di T e sia $NR_T = NR_R$ il numero di record di T. Si ordina T usando la combinazione lessicografica di tutti gli attributi.

$$C(\text{sort}(EP_T)) = C_{I/O}(\text{sort}(EP_T)) + C_{CPU}(\text{sort}(EP_T))$$

$$C_{I/O}(\text{sort}(EP_T)) = 2 \cdot EP_T \cdot (1 + \log_z(\frac{EP_T}{(Z+1) \cdot FS}))$$

3. Si scandisce T eliminando i duplicati :

$$C(\text{seq T}) = EP_T + \alpha \cdot NR_T$$

Query di join

Theta join

Assumiamo di avere 2 relazioni, chiamate rispettivamente R ed S.

$f_{F_{J(R,S)}} = \frac{|R \bowtie S|}{|R \times S|}$: fattore di selettività senza predicati locali

$f_{R,S}$: fattore di selettività della query di join con predicati locali

$ER_{q_{R,S}}(ExpectedRecords from q_{r,s}) = f_{R,S} \times NR_R \times NR_S$: numero di record risultanti dalla query $q_{R,S}$

$ER_{F_{J(R,S)}} = f_{F_{J(R,S)}} \times NR_R \times NR_S$: numero di record del risultato di puro join senza predicati locali

Equi-join

due relazioni legate da una condizione di uguaglianza

$$f_{P(R.S=R.R)} = \frac{1}{\max(NK_{R.S}, NK_{R.R})}$$

caso 1 : equi-join su primary e foreign key

con $NK_{R.J}$ e $NK_{S.J}$ cardinalità degli attributi R.J e S.J

$$f_{p(R.J=S.J)} = \frac{ER}{NR_R \times NR_S} = \frac{NR_S}{NR_R \times NR_S} = \frac{1}{NK_{R.J}}$$

$ER_{F_{J(R,S)}} = NR_S$ (R.J è PK mentre S.J è FK)

caso 2 : equi-join m : n (la stima non è su PK ne su SK)

$$f_{p(R.J=S.J)} = \frac{1}{NK_{R.J}}$$

$$ER_{F_{J(R,S)}} = f_{p(R.J=S.J)} \times NR_R \times NR_S$$