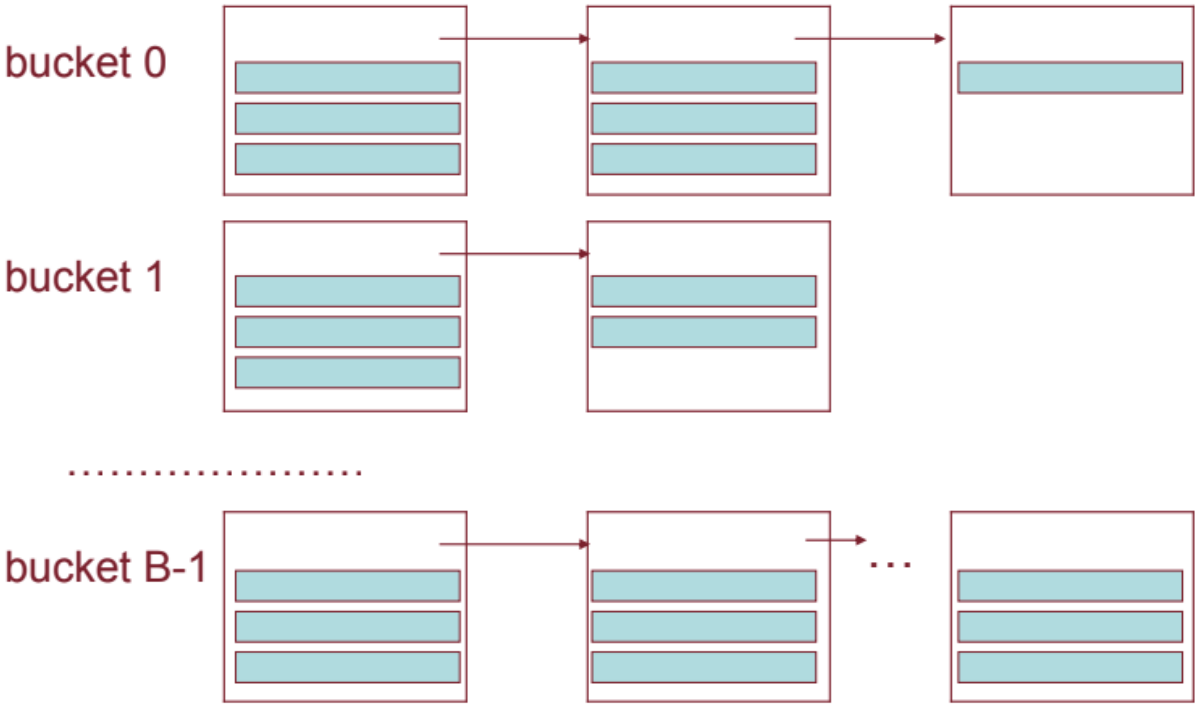
File hash - caratteristiche ed esercizi

Introduction

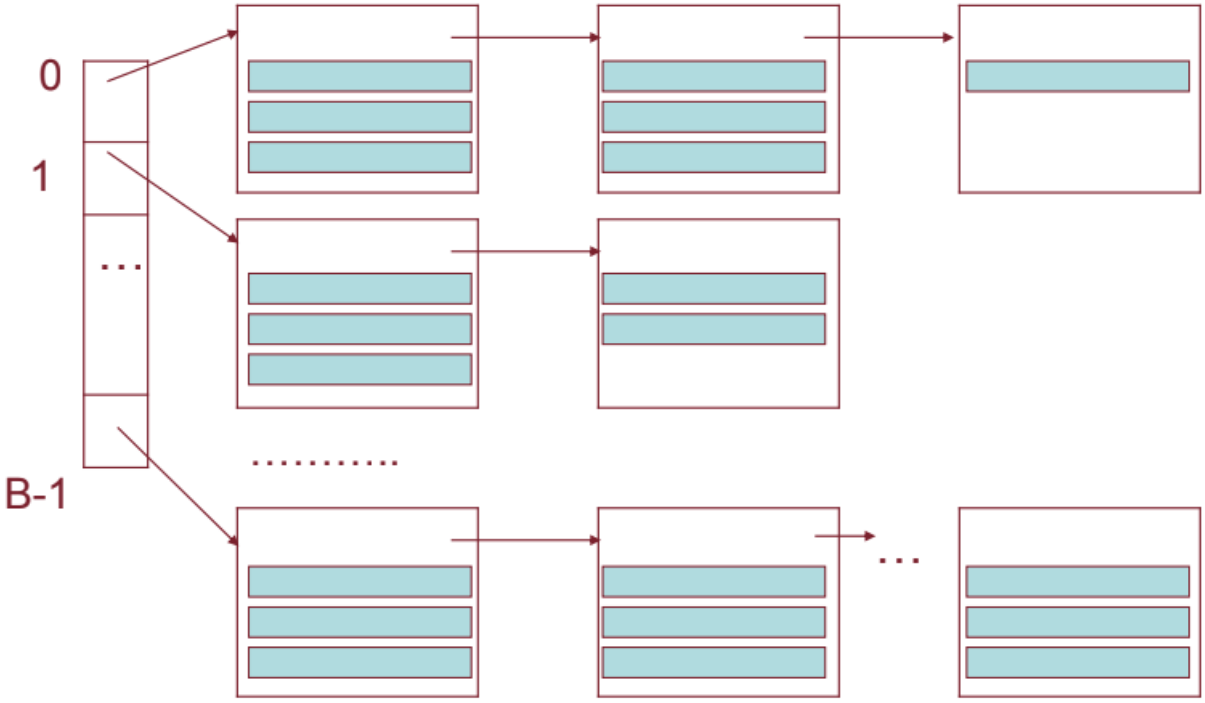
La particolarità dei file hash è il fatto che il file è suddiviso in **bucket** numerati da 0 a $B-1$. Ciascun bucket è costituito da **uno o più blocchi** collegati mediante puntatori ed è organizzato come un heap

Bucket



Bucket directory

L'accesso ai bucket avviene attraverso la **bucket directory** che contiene B elementi. L'i-esimo elemento contiene l'indirizzo del primo blocco dell'i-esimo bucket (**bucket header**)



Funzione di hash

Dato un valore v per la chiave, il numero del bucket in cui deve trovarsi un record con chiave v è calcolato mediante una funzione che prende il nome di **funzione di hash**

Una funzione hash per essere “buona” deve ripartire uniformemente i record nei bucket, cioè al variare del valore della chiave deve assumere con la “stessa” probabilità uno dei valori compresi tra 0 e $B-1$.
In generale, una funzione hash trasforma la chiave in un intero, divide questo intero per B e fornisce il resto della divisione come numero del bucket in cui deve trovarsi il record con quel valore della chiave

Esempio di funzione hash

1	0	0	1	0	1	1	1	1	0	1	0

v

1. trattare il valore v della chiave come una sequenza di bit
2. suddividere tale sequenza in gruppi di bit di uguale lunghezza e sommare tali gruppi trattandoli come interi

1	0	0	1	0	1	1	1	1	0	1	0

$$9 + 7 + 10 = 26$$

3. dividere il risultato per il numero dei bucket (cioè per B) e prendere il resto della divisione come numero del bucket in cui deve trovarsi il record con chiave v
   ES: se $B = 3$ allora il record con chiave v deve trovarsi nel bucket 2 in quanto $26 = 3 * 8 + 2$

Operazioni

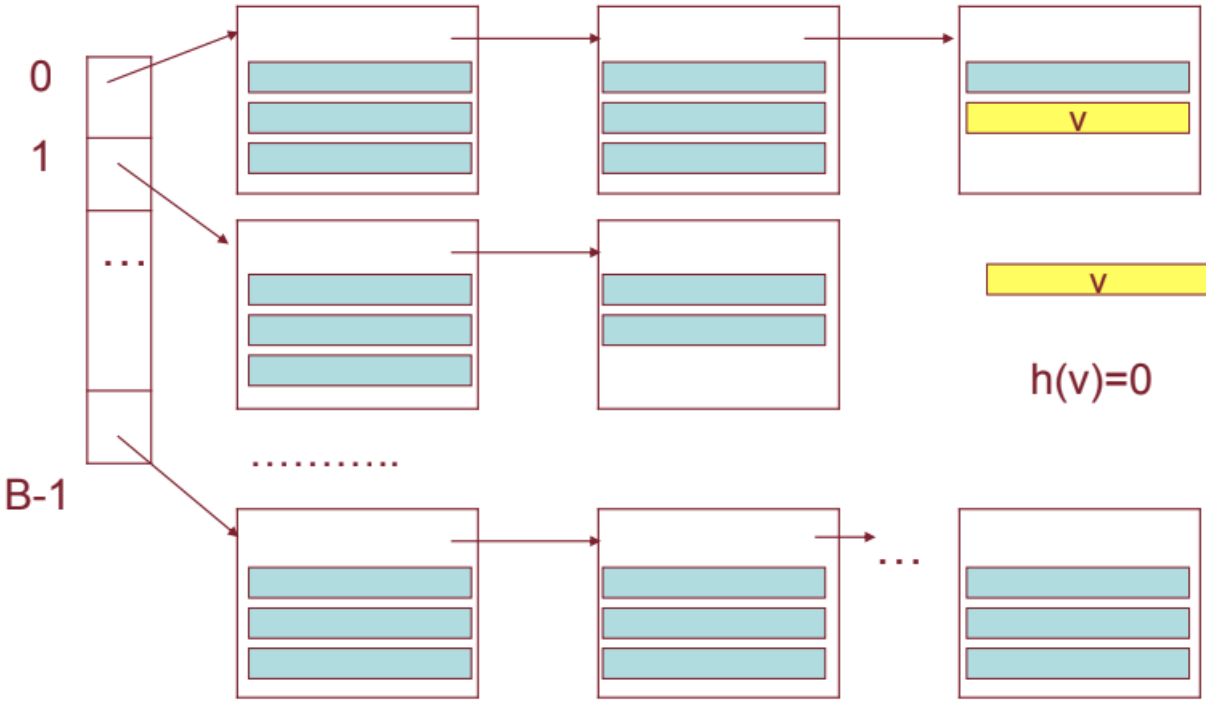
Una qualsiasi operazione (ricerca, inserimento, cancellazione, modifica) su un file hash richiede:

- la valutazione di $h(v)$ per individuare il bucket
- esecuzione dell'operazione sul bucket che è organizzato come un heap
  Poiché l'inserimento di un record viene effettuato sull'ultimo blocco del bucket è opportuno che la bucket directory contenga anche, per ogni bucket, il puntatore all'ultimo record del bucket

Costo operazioni

Pertanto se la funzione hash distribuisce uniformemente i record nei bucket allora ogni bucket è costituito da $\frac{n}{B}$ blocchi e quindi il costo richiesto di un'operazione è approssimativamente $\frac{1}{B\text{-esimo}}$ del costo della stessa operazione se il file fosse organizzato come heap

Inserimento



Considerazioni

Da quanto detto appare evidente che quanti più sono i bucket tanto è più basso il costo di ogni operazione. D'altra parte limitazioni al numero di bucket derivano dalle seguenti considerazioni:

- ogni bucket deve avere almeno un blocco
- se le dimensioni della bucket directory sono tali che non può essere mantenuta in memoria principale durante l'utilizzo del file, ulteriori accessi sono necessari per leggere i blocchi dalla bucket directory

---

Esempi

Negli esempi che seguono, così come negli esercizi di esame, a meno che non venga specificato diversamente assumeremo sempre che:

- ogni record deve essere contenuto completamente in un blocco (non possiamo avere record a cavallo di blocchi)
- i blocchi vengono allocati per intero (non possiamo allocare frazioni di blocco)

Esempio 1 >

Supponiamo di avere un file di 250.000 record. Ogni record occupa 300 byte, di cui 75 per il campo chiave. Ogni blocco contiene 1024 byte. Un puntatore a blocco occupa 4 byte

1. Se usiamo una organizzazione hash con 1200 bucket, quanti blocchi occorrono per la bucket directory?
2. Quanti blocchi occorrono per i bucket, assumendo una distribuzione uniforme dei record nei bucket?
3. Assumendo ancora che tutti i bucket contengano il numero medio di record, qual è il numero medio di accessi a blocco per ricercare un record che sia presente nel file?
4. Quanti bucket dovremmo creare per avere invece un numero medio di accessi a blocco inferiore o al massimo uguale a 10, assumendo comunque una distribuzione uniforme dei record nei bucket?

Abbiamo i seguenti dati:

- il file contiene 250.000 record $\rightarrow NR = 250.000$

- ogni record occupa 300 byte $\rightarrow R = 300$
- il campo chiave occupa 75 byte $\rightarrow K = 75$
- ogni blocco contiene 1024 byte $\rightarrow CB = 1024$
- un puntatore a blocco occupa 4 byte $\rightarrow P = 4$

> **⚠ Warning**
>
> Un calcolo del tipo $\frac{NR \cdot R}{CB}$ per calcolare l'occupazione totale è sbagliato per tre motivi:
>
> - avremmo record a cavallo di blocchi (se la taglia non è divisibile)
> - avremmo blocchi a cavallo di bucket (se gli ultimi blocchi del bucket non sono riempiti per intero)
> - mancano i puntatori al prossimo blocco nel bucket

1

Indichiamo con B il numero di bucket e con BD il numero di blocchi per la bucket directory. La bucket directory è essenzialmente un array di puntatori indicizzato da 0 a $B - 1$

Vediamo prima quanti puntatori entrano in un blocco (prendiamo la parte intera inferiore perché assumiamo che i record siano contenuti interamente nel blocco)

$$PB = \left\lfloor \frac{CB}{P} \right\rfloor = \left\lfloor \frac{1024}{4} \right\rfloor = 256$$

Ci occorreranno (prendiamo la parte intera superiore perché, non essendo stato specificato diversamente dall'esercizio, i blocchi vengono allocati interamente, e quindi la frazione di blocco va arrotondata ad un blocco intero)

$$BD = \left\lceil \frac{1200}{256} \right\rceil = \lceil 4.69 \rceil = 5$$

> **ⓘ Info**
>
> Se viene chiesto che nella bucket directory venga memorizzato anche il puntatore all'ultimo blocco del bucket occorrere considerare coppie intere di puntatori (non possiamo spezzare in due blocchi la coppia di puntatori per un blocco)

$$PB = \left\lfloor \frac{CB}{2P} \right\rfloor$$

2

Abbiamo record a lunghezza fissa, quindi supponiamo di non avere un direttorio di record all'inizio del blocco (tutto lo spazio è occupato dai dati). Serve però un puntatore per ogni blocco per linkare i blocchi dello stesso bucket. In un blocco dobbiamo quindi memorizzare il maggior numero possibile di record e in più un puntatore per un eventuale prossimo blocco nel bucket.
Se indichiamo con M il massimo numero di record memorizzabili in un blocco, avremo $M \cdot R + P \leq CB$, cioè $300M + 4 \leq 1024$, quindi $M \leq \frac{1020}{300} = 3.4$
M deve essere intero, perché non essendo stato detto altrimenti nella traccia, assumiamo che i record non possano trovarsi a cavallo di due o più blocchi, quindi assumiamo $M = 3$

In alternativa possiamo prima sottrarre la taglia del puntatore dallo spazio utile e poi prendere la parte intera inferiore della divisione dello spazio rimanente per la taglia dei record

$$M = \left\lfloor \frac{CB - P}{R} \right\rfloor = \left\lfloor \frac{1020}{300} \right\rfloor = \lfloor 3.4 \rfloor = 3$$

> **Info**
>
> Si potrebbe chiedere che ogni blocco abbia anche un puntatore al blocco precedente quindi $M \cdot R + 2P \leq CB$ oppure $M = \lfloor \frac{CB-2P}{R} \rfloor$

Se la distribuzione dei record nei bucket è uniforme, indicando con RB il numero di record in un bucket, avremo

$$RB = \left\lceil \frac{NR}{B} \right\rceil = \left\lceil \frac{250.000}{1200} = \right\rceil = \lceil 208.3 \rceil = 209$$

record per ogni bucket (prediamo la parte intera superiore perché i record devono essere inseriti tutti, quindi la frazione di record va considerata per non tralasciare alla fine di una parte dei record stessi)

Indicando con NB il numero di blocchi per ogni bucket, occorrono quindi:

$$NB = \left\lceil \frac{RB}{M} \right\rceil = \left\lceil \frac{209}{3} \right\rceil = 70$$

blocchi per ogni bucket; indicando con BB il numero complessivo di blocchi per il file hash avremo

$$BB = NB \cdot B = 70 \cdot 1200 = 84.000$$

blocchi

3

Se la distribuzione dei record nei bucket è uniforme, in un bucket avremo, come detto, $NB = \lceil \frac{RB}{M} \rceil = 70$ blocchi per bucket. Poiché la ricerca avviene solo sul bucket individuato in base al risultato dell'applicazione della funzione hash alla chiave del record, avremo un numero di accessi pari a quello che si avrebbe su un heap della stessa dimensione del bucket (cioè $\frac{1}{B}$ rispetto alla dimensione originale).
In media accederemo alla metà di questi blocchi, quindi indichiamo con MA il numero medio di accessi

$$MA = \left\lceil \frac{NB}{2} \right\rceil = \left\lceil \frac{70}{2} \right\rceil = 35$$

A questi occorrerà aggiungerne 1 se il blocco della bucket directory relativo al bucket in cui si trova il record non si trova già in memoria principale

4

Per avere un numero di accessi a blocco inferiore o al massimo uguale a 10, riscriviamo l'espressione di MA in modo che vi compaia esplicitamente il numero di bucket B, e tralasciando per semplicità gli arrotondamenti che abbiamo effettuato via via nei calcoli tranne l'ultimo. Avremo

$$MA = \left\lceil \frac{NB}{2} \right\rceil = \lceil \frac{\frac{RB}{m}}{2} \rceil = \lceil \frac{\frac{\frac{NR}{B}}{M}}{2} \rceil = \left\lceil \frac{NR}{2(B \cdot M)} \right\rceil$$

Vogliamo calcolare quindi B in modo tale che

$$\left\lceil \frac{NR}{2(B \cdot M)} \right\rceil \implies B \geq \frac{NR}{20M} \implies B \geq \frac{250.000}{20 \cdot_3} = 4167$$

Verifichiamo infatti che in questo caso avremo

$$RB = \left\lceil \frac{NR}{B} \right\rceil = \left\lceil \frac{250.000}{4167} \right\rceil = 60$$

record per ogni bucket, quindi

$$NB = \left\lfloor \frac{RB}{M} \right\rfloor = 20$$

record per bucket, e infine

$$MA = \left\lceil \frac{NB}{2} \right\rceil = 10$$

accessi a blocco

Si poteva anche ragionare in un altro modo. Siccome $MA = \lceil \frac{NB}{2} \rceil$, per avere $MA \leq 10$ dobbiamo fare in modo che $NB \leq 20$ (ricordiamo che NB è il numero di blocchi in un bucket).
Dobbiamo allora avere $RB = M \cdot NB \leq M \cdot 20$, cioè nel nostro caso $RB \leq 60$ (ricordiamo che RB è il numero di record per bucket).
Per avere un numero di record per bucket inferiore a 60, deve essere $\frac{NR}{B} \leq 60$, e quindi $B \geq \frac{250.000}{60}$ ottenendo lo stesso risultato

Esempio 2 ›

Supponiamo di avere un file di 780.000 record. Ogni record occupa 250. Ogni blocco contiene 1024 byte. Un puntatore a blocco occupa 4 byte. Usiamo una organizzazione hash con 2500 bucket

1. Quanti blocchi dobbiamo utilizzare complessivamente per la bucket directory e per i bucket, assumendo una distribuzione uniforme dei record nei bucket
2. Quanti blocchi dobbiamo utilizzare complessivamente per i bucket, assumendo che il 30% dei record sia distribuito in modo uniforme su 1000 bucket, e che il restante 70% dei record sia distribuito in modo uniforme sui 1500 bucket rimanenti

Abbiamo i seguenti dati:

- Numero di record $\rightarrow NR = 780.000$
- Taglia record $\rightarrow R = 250$ byte
- Taglia puntatore $\rightarrow P = 4$ byte
- Capacità blocco $\rightarrow CB = 1024$ byte
- Numero bucket $\rightarrow B = 2500$

1

Calcoliamo innanzitutto quanti puntatori entrano in ogni blocco della bucket directory

$$PB = \left\lfloor \frac{CB}{P} \right\rfloor = \left\lfloor \frac{1024}{4} \right\rfloor = 256$$

Calcoliamo quindi quanti blocchi occorrono per la bucket directory, cioè per memorizzare 2500 puntatori

$$BD = \left\lceil \frac{B}{PB} \right\rceil = \left\lceil \frac{2500}{256} \right\rceil = \lceil 9.7 \rceil = 10$$

Assumendo una distribuzione uniforme, dobbiamo prima di tutto calcolare quanti record devono essere memorizzati in ogni bucket

$$RB = \left\lceil \frac{NR}{B} \right\rceil = \left\lceil \frac{780.000}{2500} \right\rceil = \lceil 312 \rceil = 312$$

Vediamo ora quanti record possono essere memorizzati in un blocco, tenendo conto del fatto che ogni blocco di un bucket deve contenere anche un puntatore al blocco successivo

$$RBL = \left\lfloor \frac{CB - P}{R} \right\rfloor = \left\lfloor \frac{1020}{250} \right\rfloor = \lfloor 4.08 \rfloor = 4$$

Calcoliamo infine quanti blocchi occorrono per ogni bucket

$$BB = \left\lceil \frac{RB}{RBL} \right\rceil = \left\lceil \frac{312}{4} \right\rceil = \left\lceil \frac{NR}{B} \right\rceil = \lceil 78 \rceil = 78$$

Complessivamente ci occorre un numero di blocchi pari a
$BD + BB \cdot B = 10 + 78 \cdot 2500 = 195.010$

2

I calcoli per la bucket directory e per il numero di record che possono essere memorizzati in un blocco rimangono validi anche in questo caso

Cambia la distribuzione dei record nei bucket, e quindi il numero di blocchi che occorrono per ogni bucket. Il 30% dei record è distribuito uniformemente su 1000 bucket, il rimanente 70% dei record è distribuito uniformemente su 1500 bucket.

Quindi il numero di record che va distribuito uniformemente su 1000 bucket è

$$N_{1000} = \frac{780.000 \cdot 30}{100} = 234.000$$

Per ogni bucket dei 1000 avremo quindi $RB_{1000} = \lceil \frac{N_{1000}}{1000} \rceil = 234$ record
Quindi per ognuno di questi 1000 bucket occorrono
$B_{1000} = \lceil \frac{RB_{1000}}{RBL} \rceil = \lceil \frac{234}{4} \rceil = \lceil 58.5 \rceil = 59$ blocchi

Il numero di record che va distribuito uniformemente su 1500 bucket è

$$N_{1500} = N - N_{1000} = 780.000 - 234.000 = 546.000$$

Per ogni bucket dei 1500 avremo quindi $RB_{1500} = \lceil \frac{N_{1500}}{1500} \rceil = 364$ record
Quindi per ognuno di questi 1500 bucket occorrono quindi
$B_{1500} = \lceil \frac{RB_{1500}}{RBL} \rceil = \lceil \frac{364}{4} \rceil = \lceil 91 \rceil = 91$ blocchi

In totale avremo $B_{1000} \cdot 1000 + B_{1500} \cdot 1500 = 59 \cdot 1000 + 91 \cdot 1500 = 195.500$ blocchi per bucket

Info

Notare che a parità di spazio totale occupato, nella prima parte del file ho un tempo di ricerca medio che è quasi la metà della seconda parte ($\lceil \frac{59}{2} \rceil$ contro $\lceil \frac{91}{2} \rceil$)