

→ long pin

Necesitamos un mecanismo de acceso a registros con una lectura solamente

- Secuencia: $N/2$ accesos promedio
- Ordenado: $\log_2 N$
- Árboles: 3 o 4 accesos

Clave Primarias → características

- No se repiten
- El resto de las claves actúan a través de ella
- Cuando se aprenda a modelar, tendrán más características que las hacen especiales

FOD - CLASE 7

UNAF - Facultad de Informática

Cosas q' incluyen en eficiencia

- Tamaño de nodo → + de una clave primaria puede dar = dir. a la clave
- Claves secundarias

Collision → cuando se asigna = dir. a la clave a 2 Claves primarias.

→ si hay nodo de pin 2 y hay 2, está en collision pero no hay drama. si es un tercer key of.

- método de tratamiento de OL.

función de hash → cómo hay q. averiguar dónde va pin.

Algoritmos simples de dispersión

- Condiciones
- Repartir registros en forma uniforme
- Aleatoria (las claves son independientes, no influyen una sobre la otra)

Tres pasos

- Representar la llave en forma numérica (en caso que no lo sea)
- Aplicar la función
- Relacionar el número resultante con el espacio disponible



→ ¿qué me da 300 de rango de 300 lugares donde lo meto?

↓
Se aplica muy difícil
↓
+ o - uniforme.

Peor situación,
collision en todas las claves.

1. Función de Hashing

$$f(x) = h \quad 1 + (143)^2 = 0.296, 3493, 3909$$

$$1493 \cdot 0,2 = 298$$

2000 pos

Claves secundarias

→ Clave se multiplica x sí misma y se toman dígitos centrales del 2. Se ajusta al rango disponible.

• A partir de f. de hashing consigo dir. a la clave.

• El espacio de no 1...4 memoria mediante el hash y dir. a la clave.

• A partir de f. de hashing conigo direl.

• En dispersión NO hay remisión aparente p/ lado y direl.

• 2 llaves para misma = direl.

Tamaño de compartimiento

• A + tamaño, NO - colisión

- overflow, → dir. m. req. hay de haber colisión
- Se busca + levo de no del nodo (1 solo acceso a mem igual)
- + Atragarramiento (espacio reservado al peso)

• Para almacenar + de un reg.

Definición de empujamiento → DE 100% → = cant de reg. que espacio

• Proporción de espacio del archivo asignado p/ para almacenar reg

$$DE = \frac{\text{nº de reg del arch}}{\text{capacidad total del archivo}} \rightarrow \text{Rango q' sabemos a priori}$$

• DE menor

- Lugar muy grande p. para arch.
- - OF
- + desperdicio de espacio

Estimación del overflow

• Variable no de nodos p , C capacidad de nodo

• R cant de reg del arch

$$OE = \frac{R}{C \cdot N}$$

• Probabilidad de q' un nodo tenga $\leq R$ reg → distribución de Poisson

$$P(\text{cant de reg en un nodo}) = P(I) = \frac{R!}{I! \cdot (R-I)!} \cdot \left(\frac{1}{N}\right)^I \cdot \left(1 - \frac{1}{N}\right)^{R-I}$$

Análisis numéricos de Hashing

• En general si hay n direcciones, entonces el # esperado de direcciones con l registros asignados es $N \cdot P(l)$.

• Las colisiones aumentan con el archivo más "lleno"

• Ej: $N = 10000$ $R = 10000$ $DE = 1$ 100% → $C = 1$

$P(0) = 0.3679$		3679
$P(1) = 0.3679$	$\cdot 10000$	3679
$P(2) = 0.1839$	→ 1 en 2	1839
$P(3) = 0.0613$	→ 2 en 3	613

qué significa?

→ DE 1 en 10000 direcciones 3679 q' van a quedar libres

DE 1 en 10000 millones reg en d.

overflow = $1839 + 2 \cdot 613 = 3065$ (alto)

Ahora supongamos que el problema es

- $R = 500$ $N = 1000$ $DE = 50\%$
 $P(0) = 0.607$ $607 \rightarrow$ vacío
 $P(1) = 0.303$ $* 1000$ $303 \rightarrow$ 2 personas
 saturación = $N * [1 * P(2) + 2 * P(3) + 3 * P(4) + 4 * P(5)] = 107$

• Saturación menor

densidad	overflow
10%	4.8%
50%	21.4% $(107/500)$
100%	36.8%

- los números bajos de overflow (baja densidad) \rightarrow muchas cubetas libres

DE	1	2	5	10	100
10%	4.8	0.6	0.0	0.0	0.0
20%	9.4	2.2	0.1	0.0	0.0
30%	13.6	4.5	0.4	0.0	0.0
40%	17.6	7.3	1.1	0.1	0.0
50%	21.3	10.4	2.5	0.4	0.0
60%	24.8	13.7	4.5	1.3	0.0
70%	28.1	17.0	7.1	2.9	0.0
75%	29.6	18.7	8.6	4.0	0.0
80%	31.2	20.4	10.3	5.3	0.1
90%	34.1	23.8	13.8	8.9	0.8
100%	36.8	27.1	17.6	12.5	4.0

Peor or., + Apiden un menor de otros.

Método de saturación de desbordamiento

- Saturación progresiva \rightarrow Donde no hay lugar, busca a partir de ahí, pero además donde hay lugar
 \rightarrow Si no hay lugar busca desde el principio
 \rightarrow Marca de q' hubo algo

1	L
2	
3	a
4	b
5	c

$$f(a) = 3 \rightarrow 1 \text{ acceso}$$

$$f(b) = 3 \rightarrow \text{está ocupado, en a, y} \rightarrow 2 \text{ acc.$$

$$f(c) = 3 \rightarrow 3 \text{ acc.}$$

$$f(L) = 4 \rightarrow 3 \text{ accesos.}$$

1	L
2	
3	a
4	a
5	c

$$f(a) = 3$$

$$f(b) = 3$$

$$f(c) = 3$$

$$f(L) = 4$$

-b \rightarrow Marca de borrado

-D \rightarrow pero como vacío, no como hay m. de borrado

• Saturación progresiva enlazada

Regresa al punto al que se enlaza.

Problema si ocupa clave base con otro, de otro.

1	L	1
2		
3	a	5
4		

$$f(a) = 3$$

$$f(c) = 3$$

3	a	5
4		
5	c	1

$$f(c) = 3$$

$$f(l) = 4$$

• Áreas desbordadas separadas.

1	1	1
2		
3	B	2
4		
5		

BASE

Áreas de desbordar

1	C	3
2	e	4
3	R	-1
4	N	-1
5		

$$f(A) = 1$$

$$f(B) = 3$$

$$f(C) = 1$$

$$f(e) = 3$$

$$f(R) = 1$$

$$f(N) = 3$$

• Doble dispersión

Don P. de hashing

Algun forma de OF de la direct. Base.

1	D
2	
3	A
4	B
5	C
6	E
7	

$$f_1(A) = 3$$

$$f_1(B) = 4$$

$$f_1(C) = 3 / f_2(C) = 2 \rightarrow \text{distancia 1 y otra columna por el 2}$$

$$f_1(E) = 3 / f_2(E) = 4$$

$$f_1(D) = 5 / f_2(D) = 3$$

Dispersión, entonces se ha dado cerquita.