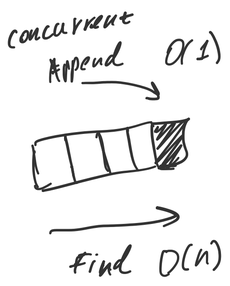
среда, 26 октября 2022 г. 20:03

Indexing

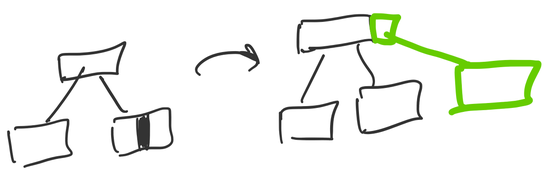


1) Append to file

2) In-memory hash index
record → offset
find O(1)

3) B-tree, LSM tree

B-tree, fault tolerance
- WAL
- copy-on-write
- concurrency
  latch ~ lock



Bloom Filter

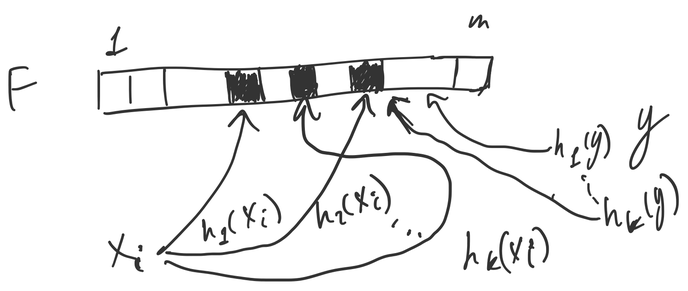
$x_i \in K, |K|$

$S = \{x_1, x_2, \dots, x_n\}$ $\quad P(x \notin S) < \varepsilon$, $P(x \in S) \geq 1 - \varepsilon$

$$\text{contain}(S, x) = \begin{cases} \text{true}, & x \in S \\ \text{false}, & x \notin S \end{cases}$$

S space $n \cdot \log |K|$

Filter space $\text{const} \cdot n = m$



$y \notin S$

$P[F[h_1(y)] = 1] = 1 - P[F[h_1(y)] = 0] =$

$= 1 - \left(1 - \frac{1}{m}\right)^{nk}$

$P[\text{Contains}(S, y)] \simeq \left(1 - \left(1 - \frac{1}{m}\right)^{nk}\right)^k < \varepsilon$

$\simeq \left(1 - \left(\left(1 + \left(\frac{-1}{m}\right)^{-m}\right)^{-\frac{nk}{m}}\right)\right)^k =$

$(n, \varepsilon) \to m$

$(n, m) \to k \quad p \to \min$

$= \left(1 - e^{-\frac{nk}{m}}\right)^k \leq \varepsilon \quad f(k) = \ln(\varepsilon)$

$f'(k) = k \cdot \frac{1}{1 - e^{-\frac{nk}{m}}} \cdot e^{-\frac{nk}{m}}\left(-\frac{n}{m}\right) + \ln(\varepsilon) \geq 0$

$k = \lambda \cdot \frac{m}{n}$

$f'(\lambda) = \ln(1 - e^{-\lambda}) + \frac{e^{-\lambda}}{1 - e^{-\lambda}} \cdot \lambda = 0$

$\lambda = \ln 2$

$\varepsilon = \left(\frac{1}{2}\right)^{\ln 2 \cdot \frac{m}{n}} = C^{\frac{m}{n}}, \quad C < 1$

! counter
to remove

space $m \cdot \log n$

Universal hash functions

$h \in H \quad h: K \to \{0, \dots, m-1\}$

$\forall k, l \in K, \; k \neq l$

$\#\{h \in H \mid h(k) = h(l)\} \leq \frac{|H|}{m}$

$$\frac{\#\{h \in H \mid h(k) = h(l)\}}{|H|} \leq \frac{1}{m}$$

$h \in H$

Example

$H_{p,m} = \{h_{(a,b)} \mid a \in \underbrace{\mathbb{Z}_p^*}_{1, \dots, p-1}, b \in \underbrace{\mathbb{Z}_p}_{0, \dots, p-1}\}$ p – prime

$h_{(a,b)}(k) = ((ak + b) \% p) \% m$

Materialized view

time \ key	I	II	III
10	1	2	0
16	4	3	1
20	0	2	1

time \ key	I	II	III
10	1	3	3
16	5	10	11
20	5	12	14

to persist data

fdatasync
fsync

Select count(*)

data

key	time	value

secondary indexes

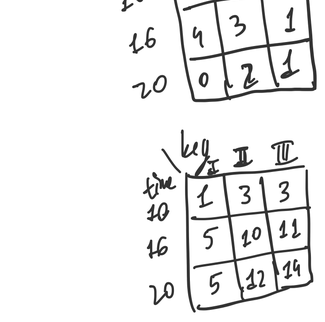
key time value

multi column index

concat VS R-tree

heap file VS cluster index

key → row



full-text index

query

document

Sphinx term → (doc_id, pos)

$tf(t, d) = \frac{n_t}{\sum_k n_k}$

$idf(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|}$

$tf\text{-}idf(t, d, D) = tf \cdot idf$