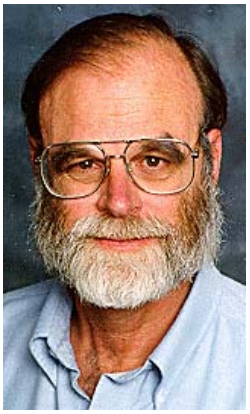


个人简介
专业打杂程序员
联系方式
新浪微博 腾讯微博

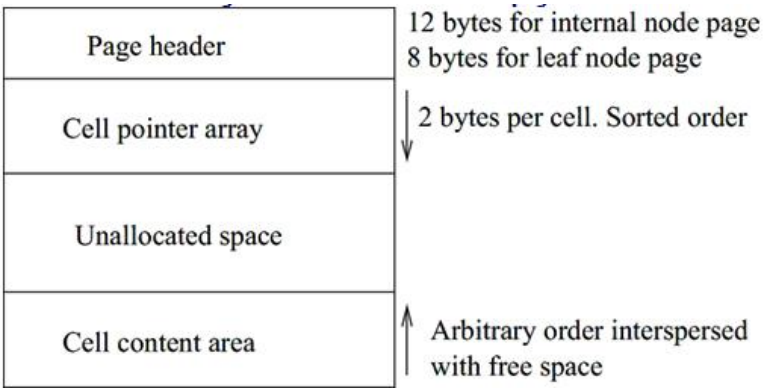
SQLite入门与分析(八)---存储模型(2)

3、页面结构（page structure）

数据库文件分成固定大小的页面。SQLite通过B+tree模型来管理所有的页面。页面(page)分三种类型：要么是tree page，或者是overflow page，或者是free page。

3.1、Tree page structure

每个tree page分成许多单元(cell)，一个单元包含一个(或部分)payload。Cell是tree page进行分配或回收的基本单位。
一个tree page分成四个部分：



(1)The page header
(2)The cell content area
(3)The cell pointer array
(4)Unallocated space
Cell 指针数组与cell content相向增长。一个page header仅包含用来管理页面的信息，它通常位于页面的开始处(但是对于数据库文件的第一个页面，它开始第100个字节处，前100个字节包含文件头信息(file header))。

Structure of tree page header

Offset	Size	Description
0	1	Flags. 1: intkey, 2: zerodata, 4: leafdata, 8: leaf
1	2	Byte offset to the first free block
3	2	Number of cells on this page
5	2	First byte of the cell content area
7	1	Number of fragmented free bytes
8	4	Right child (the Ptr(n) value). Omitted on leaves.

Flag定义页面的格式：如果leaf位被设置，则该页面是一个叶子节点，没有孩子；如果zerodata位被设置，则该页面只有关键字，而没有数据；如果intkey位设置，则关键字是整型；如果leafdata位设置，则tree只存储数据在叶子节点。另外，对于内部页面(internal page)，header在第8个字节处包含指向最右边子节点的指针。
Cell位于页面的高端，而cell 指针数组位于页面的page header之后，cell指针数组包含0个或者多个的指针。每个指针占2个字节，表示在cell content区域的cell距页面开始处的偏移。页面Cell单元的数量位于偏移3处。
由于随机的插入和删除单元，将会导致一个页面上Cell和空闲区域互相交错。Cell内容区域(cell content area)中没有使用的空间收集起来形成一个空闲块链表，这些空闲块按照它们地址的升序排列。页面头1偏移处的2个字节指向空闲块链表的头。每一个空闲块至少4个字节，每个空闲块的开始4个字节存储控制信息：头2个字节指向下一个空闲块（0意味着没有下一个空闲块了），剩余的2个字节为该空闲块的大小。由于空闲块至少为4个字节大小，所以单元内容空间中的3个字节或更小的空间（叫做fragment）不能存在于空闲块列表中。所有碎片(fragment)的总的字节数将记录在页面头偏移为7的位置(所以太碎片最多为255个字节，在它达到最大值之前，页面会被整理)。单元内容区域的第一个字节记录在页面头偏移为5的地方。这个值为单元内容区域和未使用区域的分界线。

3.2、单元格式(Structure of a cell)

单元是变长的字节串。一个单元存储一个负载(payload)，它的结构如下：

Structure of a cell	
Size	**Description**

2013年9月 (1)
2013年8月 (1)
2013年2月 (1)
2012年11月 (4)
2012年1月 (1)
2011年12月 (1)
2011年10月 (1)
2011年3月 (1)
2010年9月 (1)
2010年8月 (1)
2010年7月 (3)
2010年6月 (2)
2010年5月 (7)
2010年4月 (1)
2010年3月 (1)
2010年1月 (1)
2009年12月 (2)
2009年10月 (2)
2009年9月 (14)
2009年8月 (4)
2009年6月 (14)
2009年5月 (3)
2009年4月 (1)
2009年3月 (3)
2009年2月 (11)
2008年10月 (7)
2008年8月 (5)
2008年7月 (1)
2008年6月 (2)
2008年5月 (2)
2008年4月 (5)

kernel

kernel中文社区
LDN
The Linux Document Project
The Linux Kernel Archives

manual

cppreference
gcc manual
mysql manual

sites

Database Journal
Fedora镜象
highscalability
KFUPM ePrints
Linux docs
Linux Journal
NoSQL
SQLite

技术社区

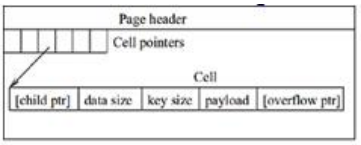
apache
CSDN
IBM-developerworks
lucene中国
nutch中国
oldlinux
oracle's forum

最新评论

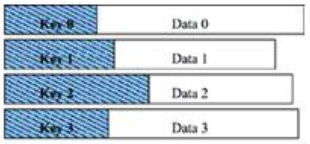
1. Re:理解MySQL——架构与概念
我试验了下.数据 5 9 10 13 18begin;select * from asf_execution where num> 5 and num 5 and INSTANCE_ID_<18 lock in share mode;会有 1.行锁 2.间隙所 [5 18)插入INSERT I.....

4	Page number of the left child. Omitted if `leaf` flag bit is set.
var(1–9)	Number of bytes of data. Omitted if the `zerodata` flag bit is set.
var(1–9)	Number of bytes of key. Or the key itself if `intkey` flag bit is set.
*	Payload
4	First page of the overflow chain. Omitted if no overflow.

对于内部页面，每个单元包含4个字节的左孩子页面指针；对于叶子页面，单元不需要孩子指针。接下来是数据的字节数，和关键字的长度，下图描述了单元格式：(a)一个单元的格式 (b)负载的结构。



(a) Structure of cell

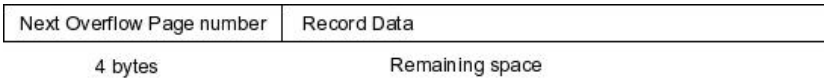


(b) Structure of payload

3.3、溢出页面

小的元组能够存储在一个页面中，但是一个大的元组可能要扩展到溢出页面，一个单元的溢出页面形成一个单独的链表。每一个溢出页面(除了最后一个页面)全部填充数据(除了最开始处的4个字节)，开始处的4个字节存储下一个溢出页面的页面号。最后一个页面甚至可以只有一个字节的数据，但是一个溢出页面绝不会存储两个单元的数据。

溢出页面的格式：



3.4、实例分析

数据库为test.db，其中有一个表和索引如下：

```
CREATE TABLE episodes( id integer primary key,name text, cid int);
CREATE INDEX name_idx on episodes(name);
```

3.4.1、叶子页面格式分析：

Episodes表的根页面为第2个页面(此时episodes表只上一个页面)，表中的数据如下：

```
sqlite> select * from episodes;
1|Cinnamon Babka2|
2|Mackinaw Peaches|1
3|Mackinaw Peaches|1
4|cat|1
5|cat|1
6|cat|1
7|cat|1
8|cat|1
9|cat|1
10|cat|1
11|cat|1
12|cat|1
13|cat2|40
14|hustcat|5
15|gloriazzz|41
16|eustcat|5
17|xloriazzz|41
```

下面为2号页面页面头(开始的8个字节)：

```
000400h: 0D 00 00 00 11 03 1C 00 03 EB 03 D4 03 BD 03 B3 ; ........????
000410h: 03 A9 03 9F 03 95 03 8B 03 81 03 77 03 6D 03 63 ; ?????w.m.c
000420h: 03 58 03 4A 03 3A 03 2C 03 1C 00 00 00 00 00 00 ; .X.J.:,........
```

Offset	Size	值 及含义

--麒麟飞
2. Re:理解MySQL——架构与概念
例1-5
insert into t(i) values(1);
这句话应该是可以插入的.
不会被阻塞
--麒麟飞
3. Re:理解MySQL——架构与概念
注：SELECT ... FOR UPDATE仅在自动提交关闭(即手动提交)时才会对元组加锁，而在自动提交时，符合条件的元组不会被加锁。

这个是错误的.自动提交的,也会尝试获取排它锁.
你可以试验下.
--麒麟飞
4. Re:浅谈mysql的两阶段提交协议
YY哥 偶像啊!细腻文笔 配有说服力的代码和图 我崇拜你 !!!
之前sqlite的深入分析帮了我大忙..现在做mysql相关 有来你的博客找东西 哈哈哈!!
--hark.perfe
5. Re:(i++)+(i++)与(++i)+(++i)
@arrowcat
这类语句本身没什么意义，但是楼主思考的角度让我豁然开朗。
--HJWAJ

0	1	0x0D: 1: intkey, 2: zerodata, 4: leafdata, 8: leaf
1	2	0x0000:第一个空闲块的偏移为0
3	2	0x0011:页面的单元数为17
5	2	0x031C:单元内容区的第一个字节的偏移(距页面起始位置)
7	1	0x00：碎片字节数
8	4	Right child (the Ptr(n) value). Omitted on leaves.

来看第2个页面的数据（0x400——0x7ff）：

```
000710h: 00 00 00 00 00 00 00 00 00 00 00 00 0E 11 04 00 ; ................
000720h: 1F 01 78 6C 6F 72 69 61 7A 7A 7A 29 0C 10 04 00 ; ..xloriazzz)....
000730h: 1B 01 65 75 73 74 63 61 74 05 0E 0F 04 00 1F 01 ; ..eustcat.......
000740h: 67 6C 6F 72 69 61 7A 7A 7A 29 0C 0E 04 00 1B 01 ; gloriazzz)......
000750h: 68 75 73 74 63 61 74 05 09 0D 04 00 15 01 63 61 ; hustcat.......ca
000760h: 74 32 28 08 0C 04 00 13 01 63 61 74 01 08 0B 04 ; t2(......cat....
000770h: 00 13 01 63 61 74 01 08 0A 04 00 13 01 63 61 74 ; ...cat.......cat
000780h: 01 08 09 04 00 13 01 63 61 74 01 08 08 04 00 13 ; .......cat......
000790h: 01 63 61 74 01 08 07 04 00 13 01 63 61 74 01 08 ; .cat.......cat..
0007a0h: 06 04 00 13 01 63 61 74 01 08 05 04 00 13 01 63 ; .....cat.......c
0007b0h: 61 74 01 08 04 04 00 13 01 63 61 74 01 15 03 04 ; at.......cat....
0007c0h: 00 2D 01 4D 61 63 6B 69 6E 61 77 20 50 65 61 63 ; .-.Mackinaw Peac
0007d0h: 68 65 73 01 15 02 04 00 2D 01 4D 61 63 6B 69 6E ; hes.....-.Mackin
0007e0h: 61 77 20 50 65 61 63 68 65 73 01 13 01 04 00 2B ; aw Peaches.....+
0007f0h: 00 43 69 6E 6E 61 6D 6F 6E 20 42 61 62 6B 61 32 ; .Cinnamon Babka2
```

页面头之后为cell指针数组，第一个cell的相对页面起始位置偏移为0x03EB，即文件的0x07EB。该单元的数据为：

```
0007e0h: 61 77 20 50 65 61 63 68 65 73 01 13 01 04 00 2B ; aw Peaches.....+
0007f0h: 00 43 69 6E 6E 61 6D 6F 6E 20 42 61 62 6B 61 32 ; .Cinnamon Babka2
```

0x13：数据的字节数，19个字节，即04 00 2B ... 61 32。
0x01：关键字的字节数，对于整型，则为关键字本身，即1。
0x04：从该字节开始为payload，即记录。0x04为记录头的大小。
0x00：NULL，id字段的值，由于关键字保存在key size中，这里为NULL
0x2B：name字段值的长度，为字符串，长度为(43-13)/2=15。
0x00：NULL，第一条记录cid的值。

（注：共有3个页面，由于篇幅所限，这里就不贴上来了）

分类: 数据库技术

YY哥
关注 - 2
粉丝 - 342
+加关注

« 上一篇：(i++)+(i++)与(++i)+(++i)
» 下一篇：计算机中的时间

posted @ 2009-06-14 16:28 YY哥 阅读(3200) 评论(4) 编辑 收藏

评论列表

#1楼 2009-06-14 22:42 neoragex2002

awesome。坚持，关注。

支持(0) 反对(0)

#2楼[楼主] 2009-06-14 22:57 YY哥

@neoragex2002
thanks

支持(0) 反对(0)

#3楼 2010-07-26 14:20 sekaii

从前一篇文章的1，怎么一下子到了本页的3了，中间落了第2章吧？

支持(0) 反对(0)

#4楼[楼主] 2010-07-26 21:29 YY哥

@sekaii
可能吧

支持(0) 反对(0)