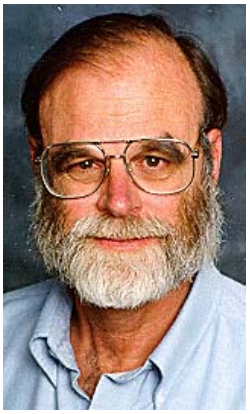

个人简介
专业打杂程序员
联系方式
新浪微博 腾讯微博

IT新闻:
美国世贸中心遗址古沉船身份判明 4分钟前
每天跑步5分钟 死亡风险降三成 6分钟前
7组数据告诉你《纸牌屋》背后的公司Netflx 如何搞定全世界 8分钟前

昵称：YY哥
园龄：7年2个月
粉丝：342
关注：2
+加关注

< 2009年9月 >

日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

搜索
找找看
谷歌搜索

常用链接
我的随笔
我的评论
我的参与
最新评论
我的标签
更多链接

随笔分类
c/c++(9)
Linux相关(24)
MySQL(11)
Others(2)
Web技术(12)
数据结构与算法(15)
数据库技术(30)
系统相关(3)
云计算与虚拟化(3)

随笔档案
2014年7月 (4)
2014年3月 (1)

SQLite入门与分析(八)---存储模型(3)

写在前面：接上一节，本节主要讨论索引页面格式，以及索引与查询优化的关系。

(1)索引页面格式
sqlite> select * from sqlite_master;
table|episodes|episodes|2|CREATE TABLE episodes(id integer primary key,name tex
t, cid int)
index|name_index|episodes|3|CREATE INDEX name_index on episodes(name)
第3个页面保存表 episodes的索引(也只占一个页面)。

```
00000800h: 0A 00 00 00 11 03 3D 00 03 EC 03 D7 03 C2 03 BA ; ......=.?????
00000810h: 03 B2 03 AA 03 A2 03 9A 03 92 03 8A 03 82 03 7A ; ???????z
00000820h: 03 71 03 4B 03 57 03 65 03 3D 00 00 00 00 00 00 ; .q.K.W.e.=......
```

前8个字节为页面头：
0x0A：leaf+zerodata，表示叶子页面，且页面中只有关键字，没有数据（即索引页面）；
0x0000：表示第一个空闲块的偏移为0；
0x0011：页面的单元数(记录数)，该页面含有17个记录；
0x030D：单元内容区的第一个字节的偏移(距页面起始位置)
0x00： 碎片字节数
接下来34个字节为17个单元（记录）的指针数组。第一个单元的偏移为0x03EC，如

```
00000b30h: 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 03 1F ; ................
00000b40h: 01 78 6C 6F 72 69 61 7A 7A 7A 11 0B 03 1B 01 65 ; .xloriazzz.....e
00000b50h: 75 73 74 63 61 74 10 0D 03 1F 01 67 6C 6F 72 69 ; ustcat.....glori
00000b60h: 61 7A 7A 7A 0F 0B 03 1B 01 68 75 73 74 63 61 74 ; azzz.....hustcat
00000b70h: 0E 08 03 15 01 63 61 74 32 0D 07 03 13 01 63 61 ; .....cat2.....ca
00000b80h: 74 0C 07 03 13 01 63 61 74 0B 07 03 13 01 63 61 ; t.....cat.....ca
00000b90h: 74 0A 07 03 13 01 63 61 74 09 07 03 13 01 63 61 ; t.....cat.....ca
00000ba0h: 74 08 07 03 13 01 63 61 74 07 07 03 13 01 63 61 ; t.....cat.....ca
00000bb0h: 74 06 07 03 13 01 63 61 74 05 07 03 13 01 63 61 ; t.....cat.....ca
00000bc0h: 74 04 14 03 2D 01 4D 61 63 6B 69 6E 61 77 20 50 ; t...-.Mackinaw P
00000bd0h: 65 61 63 68 65 73 03 14 03 2D 01 4D 61 63 6B 69 ; eaches...-.Macki
00000be0h: 6E 61 77 20 50 65 61 63 68 65 73 02 13 03 2B 01 ; naw Peaches..+.
00000bf0h: 43 69 6E 6E 61 6D 6F 6E 20 42 61 62 6B 61 32 01 ; Cinnamon Babka2.
```

来看看索引单元的格式：
0x13：数据的字节数，19个字节，从0x03开始。
0x03：记录头的字节数。
0x2B：第一个字段的长度，15个字节，该索引是对episodes表的name字段建的，其值为episodes表name字段的值。
0x01：第二个字段的长度，其值为0x01，即episodes表中的对应记录rowid的值。

(2)索引与order by
order by是查询中经常用到的，一些通用DBMS(比如DM，Mysql)都提供基于索引的形式来实现Order by。SQLite也是通过索引来实现Order by的。当字段有索引时，则直接通过索引很容易实现排序；另一方面，如果排序的字段没有索引，则以该字段为索引(这种情况下是聚集索引)建立一张临时表，再将临时表按顺序输出。来看看sqlite的实现吧。

在sqlite中，默认以rowid来建立聚集索引(对于没有整型值主键的情况)。如果主键字段为整型，则将其直接保存在rowid中，实现聚集索引；另一方面，如果主键是字符串，则对主键建立二级索引。非主键的索引都属于二级索引。
先来看看以整型ID为主键的情况：

```
///////////////以ID(rowid)为索引(即聚集索引)
sqlite> explain select * from episodes order by id;
0|Trace|0|0|0|explain select * from episodes order by id;|00|
1|Noop|0|0|0||00|
2|Goto|0|13|0||00|
3|SetNumColumns|0|3|0||00|
4|OpenRead|0|2|0||00|        ;打开表episodes,p2(=2)为其根页面
5|Rewind|0|11|0|00|          ;游标指向第一条记录
6|Rowid|0|1|0||00|           ;取出记录的rowid
7|Column|0|1|2||00|          ;取出第1列的值
8|Column|0|2|3||00|          ;取出第2列的值
9|ResultRow|1|3|0||00|       ;生成记录结果
10|Next|0|6|0||01|           ;取下一条记录
11|Close|0|0|0||00|
12|Halt|0|0|0||00|
13|Transaction|0|0|0||00|
14|VerifyCookie|0|2|0||00|
15|TableLock|0|2|0|episodes|00|
16|Goto|0|3|0||00|
```

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.....

属性有索引的情况：

```
////////////排序的实现——有索引
//算法思想：
//(1)从索引中依次读取记录(索引记录的形式如:原索引属性-rowid的键值),并取出rowid.
//(2)根据(1)中取出的rowid,在原表中查找记录,并生成记录结果.
sqlite> explain select * from episodes order by name;
0|Trace|0|0|0|explain select * from episodes order by name;|00|
1|Noop|0|0|0||00|
2|Goto|0|18|0||00|
3|SetNumColumns|0|3|0||00|
4|OpenRead|0|2|0||00|                    ;打开表,p1为表游标(0),p2为表根页面
5|SetNumColumns|0|2|0||00|
6|OpenRead|2|3|0|keyinfo(1,BINARY)|00| ;打开索引,p1为索引游标,p2为根页面
7|Rewind|2|15|1|0|00|
8|IdxRowid|2|1|0||00|                   ;从索引记录中取出rowid
9|Seek|0|1|0||00|                       ;根据rowid从表中查找记录
10|IdxRowid|2|2|0||00|
11|Column|2|0|3||00|
12|Column|0|2|4||00|
13|ResultRow|2|3|0||00|
14|Next|2|8|0||00|
15|Close|0|0|0||00|
16|Close|2|0|0||00|
17|Halt|0|0|0||00|
18|Transaction|0|0|0||00|
19|VerifyCookie|0|2|0||00|
20|TableLock|0|2|0|episodes|00|
21|Goto|0|3|0||00|
```

对于没有索引的属性排序：

```
////////////////排序的实现——没有索引
//(1)按查询属性为聚集索引建立一个临时表;
//(2)按索引顺序输出结果。
sqlite> explain select * from episodes order by cid;
0|Trace|0|0|0|explain select * from episodes order by cid;|00|
1|OpenEphemeral|1|3|0|keyinfo(1,BINARY)|00| ;p1为临时表游标,p2为临时表列数
2|Goto|0|30|0||00|
3|SetNumColumns|0|3|0||00|
4|OpenRead|0|2|0||00|       ;打开表episodes
5|Rewind|0|16|0||00|        ;游标移到表的第1条记录,p1为游标下标
6|Rowid|0|1|0||00|             ;p1为表的下标,p2指向表的记录
7|Column|0|1|2||00|         ;读取表p1(=0)的第1列
8|Column|0|2|3||00|         ;读取表p1(=0)的第2列
9|MakeRecord|1|3|4||00|
10|SCopy|3|5|0||00|
11|Sequence|1|6|0||00|
12|Move|4|7|1||00|
13|MakeRecord|5|3|8||00|
14|IdxInsert|1|8|0||00|     ;该指令在索引中插入记录,相当于对表的Insert. p1为索引下标,即OpenEphemeral打开的临时表
15|Next|0|6|0||01|
16|Close|0|0|0||00|         ;关闭表episodes
17|SetNumColumns|0|3|0||00|
18|OpenPseudo|2|1|0||00|    ;打开临时表
19|Sort|1|28|0||00|         ;与Rewind功能类似
20|Column|1|2|4||00|
21|Integer|1|8|0||00|
22|Insert|2|4|8||00|
23|Column|2|0|1||00|
24|Column|2|1|2||00|
25|Column|2|2|3||00|
26|ResultRow|1|3|0||00|     ;输出临时记录
27|Next|1|20|0||00|
28|Close|2|0|0||00|         ;
29|Halt|0|0|0||00|
30|Transaction|0|0|0||00|
31|VerifyCookie|0|2|0||00|
32|TableLock|0|2|0|episodes|00|
33|Goto|0|3|0||00|
```

--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

阅读排行榜

1. 理解MySQL——索引与优化(77641)
2. SQLite入门与分析(一)---简介(48611)
3. 理解MySQL——复制(Replication)(26213)
4. libevent源码分析(19050)
5. SQLite入门与分析(二)---设计与概念(16978)

评论排行榜

1. (i++)+(i++)与(++i)+(++i)(40)
2. SQLite入门与分析(一)---简介(30)
3. 浅谈SQLite——实现与应用(20)
4. 一道算法题,求更好的解法(18)
5. 理解MySQL——索引与优化(16)

推荐排行榜

1. SQLite入门与分析(一)---简介(12)
2. 理解MySQL——索引与优化(12)
3. 浅谈SQLite——查询处理及优化(10)
4. 乱谈服务器编程(9)
5. libevent源码分析(6)

分类: 数据库技术

绿色通道： 好文要顶 关注我 收藏该文 与我联系

YY哥
关注 - 2
粉丝 - 342
+加关注

0 0

(请您对文章做出评价)

« 上一篇：内核随记(四)--文件系统(1)
» 下一篇：SQLite入门与分析(九)---VACUUM命令分析

posted @ 2009-09-16 20:11 YY哥 阅读(2949) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

博客园首页 博问 新闻 闪存 程序员招聘 知识库

最新IT新闻:
· 人在职场：30岁，这些事再做就丢脸了！
· 支付宝，马云手中的底牌
· 苹果新Retina MacBook Pro（2014年中）开箱图+SSD简单测试
· 网吧里玩出的世界冠军 打场游戏赚了400万
· Twitter收购深度学习创业公司Madbits
» 更多新闻...

最新知识库文章:
· 如何在网页中使用留白
· SQL/NoSQL两大阵营激辩：谁更适合大数据
· 如何获取（GET）一杯咖啡——星巴克REST案例分析
· 为什么程序员的工作效率跟他们的工资不成比例
· 我眼里的DBA
» 更多知识库文章...