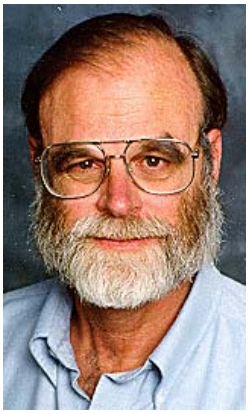
YY哥的技术随笔



个人简介
专业打杂程序员
联系方式
新浪微博 腾讯微博

< 2009年6月 >

日	一	二	三	四	五	六
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	11

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(1)

写在前面：SQLite作为嵌入式数据库，通常针对的应用的数据量相对于通常DBMS的数据量是较小的。所以它的存储模型设计得非常简单，总的来说，SQLite把一个数据文件分成若干大小相等的页面，然后以B树的形式来组织这些页面。而对于大型的数据库管理系统，比如Oracle，或者DM，存储模型要复杂得多。就拿Oracle来说吧，它对数据文件不仅从物理进行分块，而且从逻辑上进行分段，盘区和页的一个层次划分，DM也一样。不管怎么说，数据库文件要存储大量的数据，为了更好管理，查询和操作数据文件，DBMS不得不从物理上、逻辑上对数据文件的数据进行复杂的组织。本节主要讨论文件格式，下节讨论页面格式。

1、文件格式

1.1、数据库名称

应用程序通过sqlite3_open API来打开数据库，该函数的一个参数为数据库文件的名称。SQLite内部命名为main数据库(除了临时数据库和内存数据库)。SQLite对每一个数据库都创建一个独立的文件。
在SQLite内部，数据文件名不是数据库名。SQLite对应用程序的每一个连接都维护着一个单独的临时数据库(temp数据库)，临时数据库存临时对象，例如：表以及相应的索引。这些临时对象仅仅对同一个连接可见（对同一个线程，进程的其它连接是不可见的），SQLite存储临时数据库到一个单独的临时文件中，当应用程序关闭对main数据库的连接时，就删除临时文件。

1.2、数据库文件结构

除了内存数据库，SQLite把一个数据库(main和temp)都存储到一个单独的文件。

1.2.1、页面(page)
为了更好的管理和读/写数据库，SQLite把一个数据库(包括内存数据库)分成一个个固定大小的页面。页面大小的范围从512－32768（两者都包含），页面默认大小为1024个字节(1KB)，实际上，页面的上限由2个字节的有符号整数决定。整个数据库可以看成这些页面的数组，页面数组的下标为页面的编号(page number)，page number从1开始，一直到2,147,483,647 (2^31－1)。实际上，数组上界还受文件系统允许的最大文件大小决定。0号页面视为空页面(NULL page)，物理上不存在，1号页面从文件的0偏移处开始，一个页面接着下一个页面。

注：一旦数据库创建，SQLite使用编译时确定的默认的页面大小。当然，在创建第一个表之前，可以通过pragma命令改变页面大小。SQLite把该值作为元数据的一部分存储在文件中。

1.2.2、页面类型
页面(page)分四种类型：叶子页面(leaf)，内部页面(internal)，溢出页面(overflow)和空闲页面(free)。内部页面包含查询时的导航信息，叶子页面存储数据，例如元组。如果一个元组的数据太大，一个页面容纳不下，则一些数据存储在B树的页面中，余下的存储在溢出页面中。
1.2.3、文件头（file header）

作为文件开始的1号页面比较特殊，它包括100个字节的文件头。当SQLite创建文件时例初始化文件头，文件头的格式如下：

Structure of database file header

Offset	Size	Description
0	16	Header string
16	2	Page size in bytes
18	1	File format write version
19	1	File format read version
20	1	Bytes reserved at the end of each page
21	1	Max embedded payload fraction
22	1	Min embedded payload fraction
23	1	Min leaf payload fraction
24	4	File change counter
28	4	Reserved for future use
32	4	First freelist page
36	4	Number of freelist pages

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.....

40	60	15 4-byte meta values

示例数据（100个字节）：

```
53 51 4C 69 74 65 20 66 SQLite f
6F 72 6D 61 74 20 33 00 ormat 3.
04 00 01 01 00 40 20 20 .....@
00 00 00 11 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 01 00 00 00 01 ........
00 00 00 00 00 00 00 00 ........
00 00 00 01 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00
```

Header string(头字符串)：
16个字节："SQLite format 3."

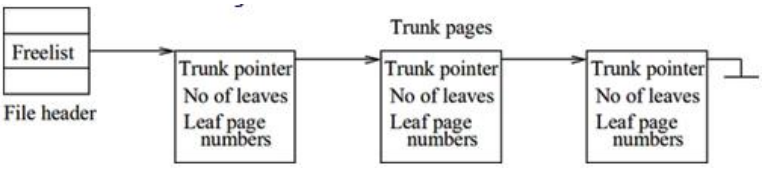
Page size：
页面大小：0x04 00，即1024
File format：
文件格式：0x01，0x01，在当前的版本都为1。

Reserved space：
保留空间：0x00，1个字节，SQLite在每个页面的末尾都会保留一定的空间，留作它用，默认为0。

Embedded payload：
max embedded payload fraction(偏移21)的值限定了B树内节点（页面）中一个元组（记录，单元）最多能够使用的空间。255意味着100%，默认值为0x40，即64（25%），这保证了一个结点（页面）至少有4个单元。如果一个单元的负载(payload，即数据量)超过最大值，则溢出的数据保存到溢出的页面，一旦SQLite分配了一个溢出页面，它会尽可能多的移动数据到溢出页面，下限为min embedded payload fraction value（偏移为22），默认的值为32，即12.5%。

min leaf payload fraction的含义与min embedded payload fraction类似，只不过是它是针对B树的叶子结点，默认值为32，即12.5%，叶子结点最大的负载为通常是100%，这不用保存。

File change counter：
文件修改计数，通常被事务使用，它由事务增加其值。该值的主要目的是数据库改变时，pager避免对缓存进行刷盘。
Freelist：
空闲页面链表，在文件头偏移32的4个字节记录着空闲页面链的第一个页面，偏移36处的4个字节为空闲页面的数量。空闲页面链表的组织形式如下：



空闲页面分为两种页面：trunk pages（主页面）和leaf pages(叶子页面)。文件头的指针指向空闲链表的第一个trunk page，每个trunk page指向多个叶子页面。
Trunk page的格式如下，从页面的起始处开始：
(1)4个字节，指向下一个trunk page的页面号；
(2)4个字节，该页面的叶子页面指针的数量；
(3)指向叶子页面的页面号，每项4个字节。

当一个页面不再使用时，SQLite把它加入空闲页面链表，并不从本地文件系统中释放掉。当添加新的数据到数据库时，SQLite就从空闲链表上取出空闲页面用来在存储数据。当空闲链表为空时，SQLite就通过本地文件系统增加新的页面，添加到数据库文件的末尾。

注：可以通过vacuum命令删除空闲链表，该命令通过把数据库中数据拷贝到临时文件，然后在事务的保护下，用临时文件中的复本覆盖原数据库文件。

Meta variables
元数据变量：从偏移为40开始，为15个4字节的元数据变量，这些元数据主要与B树和VM有关。如下：

```
** Meta values are as follows:
**    meta[0]   Schema cookie.  Changes with each schema change.
```

--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)

** meta[1] File format of schema layer.

** meta[2] Size of the page cache.

** meta[3] Use freelist if 0. Autovacuum if greater than zero.

** meta[4] Db text encoding. 1:UTF-8 2:UTF-16LE 3:UTF-16BE

** meta[5] The user cookie. Used by the application.

** meta[6]

** meta[7]

** meta[8]

** meta[9]

1.2.4、读取文件头
当应用程序调用API sqlite3_open打开数据库文件时，SQLite就会读取文件头进行数据库的初始化。

```
int sqlite3BtreeOpen(
  const char *zFilename,  /* Name of the file containing the BTree database */
  sqlite3 *pSqlite,       /* Associated database handle */
  Btree **ppBtree,        /* Pointer to new Btree object written here */
  int flags               /* Options */
){
//读取文件头
  sqlite3pager_read_fileheader(pBt->pPager, sizeof(zDbHeader), zDbHeader);
  //设置页面大小
pBt->pageSize = get2byte(&zDbHeader[16]);
//…
}
```

分类: 数据库技术

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：最大子段和——分治与动态规划
» 下一篇：(i++)+(i++)与(++i)+(++i)

posted @ 2009-06-01 21:12 YY哥 阅读(4489) 评论(2) 编辑 收藏

评论列表

#1楼 2009-06-09 20:54 广州的一学生[未注册用户]

arrowcat :
你好。
我在sqlite的源代码中看到许多PAGERTRACE，IOTRACE类似的宏，但是sqlite在运行时却没执行它们。为什么？怎样才能使sqlite执行这些代码？

博主，请你回复到我的email。谢谢！

#2楼[楼主] 2009-06-11 23:18 YY哥

@广州的一学生
那些宏是调试用的,只要在编译时定义相关的宏(比如,SQLITE_DEBUG)就可以了.

支持(0) 反对(0)

最新IT新闻:
· 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
· 这两个前亚马逊员工要把亚马逊赶出印度
» 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...