

公告

昵称: crazyYong
园龄: 8年6个月
粉丝: 507
关注: 2
+加关注

< 2021年5月 >
日 一 二 三 四 五 六
25 26 27 28 29 30 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 31 1 2 3 4 5

搜索

找我看
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

积分与排名

积分 - 521541
排名 - 774

随笔档案

2020年9月(1)
2019年5月(1)
2019年3月(4)
2019年1月(1)
2018年12月(7)
2018年12月(1)
2018年5月(1)
2018年4月(10)
2017年12月(19)
2017年11月(7)
2017年10月(36)
2017年9月(31)
2017年8月(13)
2017年7月(17)
2017年6月(3)
更多

最新评论

1. Re-Mysql加锁过程详解 (3) -关于mysql 加锁原理
中国大神的博客那里就提404了
-doubleBear
3. Re-Mysql加锁过程详解 (2) -关于mysql 加锁原理
@MMVA RR级别，假设有两条数据，事务1先执行该语句会阻塞，有两条数据，事务2接着执行这个事务插入一条数据并提交，紧接着事务1执行当前该（比如update），发现修改了3条数据，发生了幻读...
-doubleBear
4. Re-Mysql加锁过程详解 (1) -基本知识
写的太好了，结合我之前的数据仓库原理，以及索引执行计划分析，完美的验证了学到的理论，加深了记忆，赞！！！
-doubleBear
5. Re-Mysql加锁过程详解 (1) -基本知识
@世界之巅 当前锁定的话不知道是不是由于gap锁引起的阻塞，而不是死锁，可能和组合七锁似，组合七中没有提到会释放gap锁（我新手，没试过，猜的），在Repe...
-An-Illusion
5. Re-ZooKeeper学习第七期--ZooKeeper一致性原理
ZooKeeper - 一致性原理
ok正苦闷之下处于广播模式，直到leader故障被选举主之前由follower和followers，则会进入故障恢复阶段，有消息：“如果故障是一瞬间的，随即又恢复了正常，也就是说leader 有可能么可能 压报...
-lant007

Mysql加锁过程详解 (9) -innodb下的记录锁，间隙锁，next-key锁

Mysql加锁过程详解 (1) -基本知识
Mysql加锁过程详解 (2) -关于mysql加锁原理
Mysql加锁过程详解 (3) -关于mysql加锁原理
Mysql加锁过程详解 (4) -select for update/lock in share mode 对事务并发性影响
Mysql加锁过程详解 (5) -innodb 多版本开发控制原理详析
Mysql加锁过程详解 (6) -数据行级锁级别 (1)
Mysql加锁过程详解 (6) -数据行级锁级别 (2) -通过锁了解事务的4种隔离级别
Mysql加锁过程详解 (7) -初步理解MySQL的gap锁
Mysql加锁过程详解 (8) -理解innodb的锁(record, gap, Next-Key lock)
Mysql加锁过程详解 (9) -innodb 下的记录锁，间隙锁，next-key锁

你需要知道的

之前我们介绍了排他锁，其实innodb下的记录锁（也叫行锁），间隙锁，next-key锁都属于排他锁。

行锁

记录锁其实很好理解，对表中的记录加锁，叫做记录锁，简称行锁。

生活中的间隙锁

编程的思想源于生活，生活中的例子能帮助我们更好的理解一些编程中的思想。

生活中排队的情景，小明，小红，小花三个人依次站成一排，此时，如何让新来的小刚不能站在小红旁边，这时候只要将小红和前面的小明之间的空隙封锁，将小红和她后面的小花之间的空隙封锁，那么小刚就不能站到小红的旁边。

这里的小红，小明，小花，小刚就是数据库的一条记录。

他们之间的空隙也就是间隙锁，而封锁他们之间空隙的锁，叫做间隙锁。

Mysql中的间隙锁

下表中（见图一），id为主键，number字段上有非唯一索引的二级索引，有什么方式可以让该表不能再插入number=5的记录？

图一

根据上面生活中的例子，我们自然就可以想到，只要控制几个点，number=5之前不能再插入记录，number=5现有的记录之间不能再插入新的记录，number=5之后不能再插入新的记录，那么新的number=5的记录就不能被插入进来。

那么，mysql是如何控制number=5之前，之中，之后不能有新的记录插入呢（防止幻读）？

答案是用间隙锁，在RR级别下，mysql通过间隙锁可以实现锁定number=5之前的间隙，number=5记录之间的间隙，number=5之后的间隙，从而新的新的记录无法被插入进来。

间隙锁怎么划分的？

注：为了方便理解，我们规定（id=A,number=B）代表一条字段id=A,字段number=B的记录，（C，D）代表一个区间，代表C-D这个区间范围。

图一中，根据number列，我们可以分为几个区间：（无穷小，2），（2，4），（4，5），（5，5），（5,11），（11，无穷大）。

只要这些区间对应的两个边界记录中间可以插入记录，就认为区间对应的记录之间有间隙。

例如：区间（2，4）分别对应的边界记录是（id=1,number=2），（id=3，number=4），这两条记录中间可以插入（id=2,number=3）等记录，那么就认为（id=1,number=2）与（id=3，number=4）之间存在间隙。

很多人会问，那记录（id=6，number=5）与（id=8，number=5）之间有何间隙？

答案是有，（id=6，number=5）与（id=8，number=5）之间可以插入记录（id=7，number=5），因此（id=6,number=5）与（id=8,number=5）之间有间隙的。

间隙锁锁定的区域

根据检索条件向左寻找最靠近检索条件的记录值A，作为左区间，向右寻找最靠近检索条件的记录值B作为右区间，即锁定的间隙为（A，B）。

图一中，where number=5的话，那么间隙锁的区间范围为（4,11）；

间隙锁的目的是为了防止幻读，其主要通过两个方面实现这个目的：

（1）防止间隙内有新数据被插入

（2）防止已存在的数据，更新成间隙内的数据（例如防止number=3的记录通过update变成number=5)

innodb自动使用间隙锁的条件：

（1）必须在RR级别下

（2）检索条件必须有索引（没有索引的话，mysql会全表扫描，那样会锁定整张表所有的记录，包括不存在的记录，此时其他事务不能修改不能删除不能添加)

接下来，通过实际操作观察下间隙锁的作用范围

图二 表结构

图三 表结构

案例一：

.....
session 1:
start transaction ;
select * from news where number=4 for update ;

session 2:
start transaction ;
insert into news value(2,4);# (阻塞)
insert into news value(2,2);# (阻塞)
insert into news value(4,4);# (阻塞)
insert into news value(4,5);# (阻塞)
insert into news value(7,5);# (执行成功)
insert into news value(9,5);# (执行成功)
insert into news value(11,5);# (执行成功)
.....

检索条件number=4,向左取得最靠近的值2作为左区间，向右取得最靠近的5作为右区间，因此，session 1的间隙锁的范围（2，4），（4，5），如下图所示：

图二 表结构

间隙锁锁定的区间为（2，4），（4，5），即记录（id=1,number=2）和记录（id=3,number=4）之间间隙会被锁定，记录（id=3,number=4）和记录（id=6,number=5）之间间隙被锁定。

因此记录（id=2,number=4），（id=2,number=2），（id=4,number=4），（id=4,number=5）正好处在（id=3,number=4）和（id=6,number=5）之间，所以插入不了，需要等待锁的释放，而记录(id=7,number=5)，（id=9,number=5），（id=11,number=5）不在上述锁定的范围内，因此都会插入成功。

案例二：

.....
session 1:
start transaction ;
select * from news where number=13 for update ;

session 2:
start transaction ;
insert into news value(11,5);# (执行成功)
insert into news value(12,11);# (执行成功)
insert into news value(14,11);# (阻塞)
insert into news value(15,12);# (阻塞)
update news set id=14 where number=11;# (阻塞)
update news set id=11 where number=11;# (执行成功)
.....

检索条件number=13,向左取得最靠近的值11作为左区间，向右由于没有记录因此取得无穷大作为右区间，因此，session 1的间隙锁的范围（11，无穷大），如下图所示：

图二 表结构

此表中没有number=13的记录，InnoDB依然会为该记录左右两侧间隙锁，间隙锁的范围（11，无穷大）。

有人问，为更新update news set id=14 where number=11会阻塞，但是update news set id=11 where number=11却执行成功呢？

间隙锁采用在指定记录的前面和后面以及中间的间隙上加间隙锁的方式避免数据被插入，此间隙锁锁定的区域是（11，无穷大），也就是记录（id=13,number=11）之后不能再插入记录，update news set id=14 where number=11这条语句如果执行的话，将会被插入到（id=13,number=11）的后面，也就是在区间（11，无穷大）之间，由于该区间被间隙锁锁定，所以只能阻塞等待，而update news set id=11 where number=11执行后会被插入到（id=13,number=11）的记录前面，也就不在（11，无穷大）的范围内，所以无需等待，执行成功。

案例三：

.....
session 1:
start transaction ;
select * from news where number=5 for update;

session 2:
start transaction ;
insert into news value(4,4);# (阻塞)
insert into news value(4,5);# (阻塞)
insert into news value(5,5);# (阻塞)
insert into news value(7,11);# (阻塞)
insert into news value(9,12);# (执行成功)
insert into news value(12,11);# (阻塞)
update news set number=5 where id=1;# (阻塞)
update news set id=11 where number=11;# (阻塞)
update news set id=2 where number=4 ;# (执行成功)
update news set id=4 where number=4 ;# (阻塞)
.....

检索条件number=5,向左取得最靠近的值4作为左区间，向右取得11为右区间，因此，session 1的间隙锁的范围（4，5），（5，11），如下图所示：

图二 表结构

有人问，为更新insert into news value(9,12)会执行成功？间隙锁采用在指定记录的前面和后面以及中间的间隙上加间隙锁的方式避免数据被插入，（id=9,number=12）很明显在记录（13,11）的后面，因此不再锁定的间隙范围内。

为更新update news set number=5 where id=1会阻塞？

number=5的记录前面，后面包括中间都被封锁了，你这个update news set number=5 where id=1根本没法执行，因为innodb已经把你可以存放的位置都锁定了，因为只能等待。

同理，update news set id=11 where number=11由于记录（id=10,number=5）与记录（id=13,number=11）中间的间隙被封锁了，你这句sql也没法执行，必须等待，因为存放的位置被封锁了。

案例四：

.....
session 1:
start transaction;
select * from news where number>4 for update;

session 2:
start transaction;
update news set id=2 where number=4 ;# (执行成功)
update news set id=4 where number=4 ;# (阻塞)
update news set id=5 where number=5 ;# (阻塞)
insert into news value(2,3);# (执行成功)
insert into news value(null,13);# (阻塞)

检索条件number>4,向左取得最靠近的值4作为左区间，向右取无穷大，因此，session 1的间隙锁的范围（4，无穷大），如下图所示：

图二 表结构

session2中之所以有些阻塞，有些执行成功，其实就是因为插入的区域被锁定，从而阻塞。

next-key锁

next-key锁就是包含记录锁和间隙锁，即锁定一个范围，并且锁定记录本身，InnoDB默认以记录本身，InnoDB默认以记录本身是next-key 锁。

上面的案例一session 1中的sql是：select * from news where number=4 for update ；

next-key锁锁定的范围为间隙锁+记录锁，即区间（2，4），（4，5）加间隙锁，同时number=4的记录加记录锁。

from:http://www.jianshu.com/p/bf862c37c4c9

分类: mysql杂

好文推荐 关注我 收藏该文

crazyYong
关注 - 2
粉丝 - 507

+ 上一篇: Java虚拟机学习-Java虚拟机原理与源码(13-2)
+ 下一篇: eclipse安装quick test search插件，全文搜索

posted @ 2017-11-12 13:30 crazyYong 阅读(2975) 评论(7) 编辑 收藏

最新评论 刷新评论 删除页面 返回顶部

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

【推荐】云上组训 2021阿里云峰会免费门票，期待您的到来！
【推荐】大型组训、工控、仿真、CAD\GIS 50万行VC++源码免费下载！
【推荐】100个 HarmonyOS 2.0 开发Beta公测名额，限时认领！
【推荐】阿里云销量品牌榜出炉，精选爆款产品低至0.55折！
【推荐】限时秒杀！国云大数据数据，企业级云分析平台

云上创新
2021阿里云峰会
免费抢票
5.28-29 直播·直播回放

网友们：
· 欢迎各位第一时间给订书：都是我们的错
· 数据库实例 CPU 100% 引发全站故障
· 发起一个开源项目：博客引擎 fluss

阿里云 云服务器 1核2G 72元/年 限时最低价 立即购买

最新新闻：
· DOSS直播提交到IPO申请：月活3060万 去年营收19.4亿元
· 微软、余永东、李开复与扎克伯格有多远
· “反科学言论”下，电子版的真相是什么？
· Linux 5.13还原及修复明尼苏达大学的问题补丁
· 微软Android版Office终于有了黑暗模式
· 更多新闻...