请加微信1182316662 众筹更多课程1

06 | 全局锁和表锁 : 给表加个字段怎么有这么多阻碍? 2018-11-26 林晓斌



根据加赞的范围,MySQL 里面的较大致可以分成全局钱、表级铁和行铁三美。今天这篇文1 会和你分享全局锁和表级锁。而关于行锁的内容,我会留着在下一篇文章中再和你详细介绍。

到锁时的现象和其背后的原理。 全局锁

这里需要说明的是,锁的设计比较复杂,这两篇文章不会涉及锁的具体实现细节,主要介绍的是碰

前有一种做法,是通过 FTWRL 确保不会有其他线程对数据库做更新,然后对整个库做备份。注

u.account: A

M_COUYSE:

的重要数据结构。

备份状态

m.account: A 200 u_course:未备份



, single-transaction 方法只适用于所有的表使用手务引擎的库。如果有的表使用了不支持事 |摩,那么备份就只能通过 FTWRL 方法。这往往是 DBA 要求业务开发人员使用 InnoDB 替

你一定在姥姥,有了这个功能,为什么还需要 FTWRL 呢?**一数性读是好,但前提是引擎要支持这 个隔离级别。**比如,对于 MylSAM 这种不支持事务的引擎,如果备份过程中有更新,总是兄能取 到最新的数据,那么就破坏了备份的一数性。这时,我们就需要使用 FTWRL 命令了。

-是,在有些系统中,readonly 的值会被用来做其他逻辑,比如用来判断——个库是主库还是备 一意,任有認為時代,Fedoniy 的國金被用未與其他認識,氏如用來判斷一个再與土库於最新 库。因此,修改 global 安量的方式轉輸面更大,我不達议你使用。 二是,在异常於理机制上有差异。如果均方 FTWRL 命令之后由于客户端发生异常断开,那么 MySQL 会自动释放这个全局锁,整个库回到可以正常更新的状态。而将整个库设置为 readonly 之后,如果客户端发生异常,则数据库就会一直保持 readonly 状态,这样会导致整个库长时间

业务的更新不只是增删改数据(DML),还有可能是加字段等修改表结构的操作(DDL)

你也许会问,**既然要全库只读,为什么不使用 set global readonly=true 的方式呢**?确实 readonly 方式也可以让全库进入只读状态,但我还是会建议你用 FTWRL 方式,主要有两个原因:

表検的協定是 lock tables ... read/write。与FTWRL 类似,可以用 unlock tables 主动释放锁, 也可以在客户端断开的时候自动释放。需要注意,lock tables 语法除了会限制别的线程的读写外, 也限定了本线程接下来的操作对象。

一个库被全局锁上以后,你要对里面任何一个表做加字段操作,都是会被锁住的。 但是,即使没有被全局锁住,加字段也不是就能一帆风顺的,因为你还会碰到接下来我们要介绍的

举个例子,如果在某个线程 A 中执行 lock tables t1 read, t2 write; 这 读写 t2 的语句都会被阻塞。同时,线程 A 在执行 unlock tables 之前,也只能执行读 t1、读写 t2 的操作。连写 t1 都不允许,自然也不能访问其他表。

因此,在 MySQL 5.5 版本中引入了 MDL,当对-个表做增删改查操作的时候 , 加 MDL 读锁 ; 当 要对表做结构变更操作的时候,加 MDL 写锁。

• 读锁之间不互斥,因此你可以有多个线程同时对一张表增删改查。

备注:这里的实验环境是 MySQL 5.6.

MDL 写锁,因此只能被阻塞。

也会被 session C 阻塞。前面我们说了,所被锁住,等于这个表现在完全不可读写了。

请求的话,这个库的线程很快就会爆满。

transaction 參数,对应用会更友好。

tables 这样的语句,你需要追查-

你可以把你的思考和观点写在留言区里,我会在下-听,也欢迎你把这篇文章分享给更多的朋友一起阅读

说明:这篇文章没有介绍到物理备份,物理备份会有

上期的问题是关于对联合主键索引和 InnoDB 索引组织表的理解。 我直接贴 @老杨同志 的回复略作修改如下 (我修改的部分用橙色标出) :

有人掉到这个坑里:给一个小表加个字段,导致整个库挂了。 你肯定知道,给一个表加字段,或者修改字段,或者加索引,需要扫描全表的数据。在对大来操作 的时候,你肯定会特别小心,以免对线上服务造成影响。而实际上,即使显小表,操作不慎也会出 问题。我们来看一下下面的操作序列,假设表 t 是一个小表。

我们可以看到 session n A 先启动 , 这时候会对表 t 加 个 MDL 读锁。由于 session B 需要的也 MDL 读锁,因此可以正常执行。 ion C 会被 blocked,是因为 session A 的 MDL 读锁还没有释放,而 session C 需要

如果只有 session C 自己核阻塞还设什么关系,但是之后所有要在表 t 上新申请 MDL 读惯的请求 也会被 session C 阻塞。前面我们说了,所有对表的增删改直操作都需要先申请 MDL 读锁,就都

如果某个表上的查询语句频繁,而且客户端有重试机制,也就是说超时后会再起一个新 session 再

事务中的 MDL 锁,在语句执行开始时申请,但是语句结束后并不会马上释

aDB 已经合并了 AliSQL 的这个功能,所以这两个开源分支目前都支持 DDL NOWAIT/WA n 这个语法。

全局锁主要用在逻辑备份过程中。对于全部是 InnoDB 引擎的库,我建议你选择使用-single

表锁一般是在数据库引擎不支持行锁的时候才会被用到的。如果你发现你的应用程序里有 lock

-下,比较可能的情况是: • 要么是你的系统现在还在用 MyISAM 这类不支持事务的引擎 , 那要安排升级换引擎 要么是你的引擎升级了,但是代码还没升级,我见过这样的情况,最后业务开发就是把 lock tables 和 unlock tables 改成 begin 和 commit ,问题就解决了。

留一个问题吧。备份一般都会在备库上执行,你在用-single-transaction方法做逻辑 ,如果主库上的一个小表做了一个 DDL,比如给一个表上加了一列。这时候,从备库 最后,我给你留-备份的过程中 上会看到什么现象呢?

起阅读。

索引 ca 的组织是先按 c 排序,再按 a 排序,同时记录主键 -c--|-a--|-主键部分b-- (注意,这里不是 ab,而是只有 b) 213 312

緊簇索引组织顺序相当于 order by a,b ,也就是先按 a 排序,再按 b 排序,c 无序。

vangZHen 留言言简意赅 极客时间

MySQL 实战 45讲 从原理到实战,丁奇带你搞懂 MySQI

精洗留言 作者回复 Online DDL的过程是这样的: 1. 拿MDL写锁 2. 降级成MDL读锁 3. 真正做DDL

全局锁的典型使用场景是,做全库逻辑备份。也就是把整库每个表都 select 出来存成文本。 意,在备份过程中整个库完全处于只读状态。 但是让整库都只读 , 听上去就很危险 : 如果你在主库上备份,那么在备份期间都不能执行更新,业务基本上就得停摆;如果你在从库上备份,那么备份期间从库不能执行主库同步过来的 binlog,会导致主从延迟。 看来加全局锁不太好。但是细想一下,备份为什么要加锁呢?我们来看一下不加锁会有什么问题。 假设你现在要维护"极客时间"的购买系统,关注的是用户账户余额表和用户课程表。 现在发起一个逻辑备份。假设备份期间,有一个用户,他购买了一门课程,业务逻辑虽就要扣掉他 的余额,然后往已熟课程虽面加上一门课。 如果时间顺序上是先备份账户余额表 (u_account (u_course),会怎么样呢?你可以看一下这个图: account),然后用户购买,然后备份用户课程表

处于不可写状态,风险较高。

哪种方法,

构对不上,肯定是不行的。

务的引擎,那么备份就只 代 MyISAM 的原因之一。

在还没有出现更细粒度的锁的时候,表锁是最常用的处理并发的方式。 -般不使用 lock tables 命令来控制并发,毕竟锁住整个表的影响面还是太大

etadata lock)。MDL不需要显式使用,在访问 动加上。MDL 的作用是,保证读写的正确性。你可以想象一下,如果一个直询正在遍历一个表中的数据,而执行期间另一个线程对这个表结构做变更,删了一列,那么直询线程拿到的结果跟表结

同时给一个表加字段,其中一个要等另一个执行完才能开始执行。 虽然 MDL 锁是系统默认会加的,但却是你不能忽略的一个机制。比如下面这个例子,我经常看到

写锁之间 , 写锁之间是互斥的 , 用来保证变更表结构操作的安全性。因此 , 如果有两个线程要

放,而会等到整个事务提交后再释放。 基于上面的分析,我们来讨论一个问题,**如何安全地给小表加字段?** 首先我们要解决长事务,事务不提交,就会一直占着 MDL 锁。在 MySQL 的 information_schema 库的 innodb_trx 表中,你可以直到当前执行中的事更的表例好有长事务在执行,要考虑先暂停 DDL,或者 kill 掉这个长事务。 但考虑一下这个场景。如果你要变更的表是 个热点表,虽然数据量不大,但是上面的请求很频 而你不得不加个字段,你该怎么做呢? 这时候 kill 可能未必管用,因为新的南求马上就来了。比较理想的机制是,在 alter table 语句里面 设定等待时间,如果在这个指定的等待时间里面能够拿到 MDL 写锁最好,拿不到也不要阻塞后面 的业务语句,先放弃。之后开发人员或者 DBA 再通过重试命令重复这个过程。 小结 今天,我跟你介绍了 MySQL 的全局锁和表级锁。

143d 213d 222d 234d

314 423 这个跟索引 c 的数据

主键 a , b 的

上期问题时间

-a--|-b--|-c--|-d--123d

新。

所以,结论是 ca 可以去掉,cb 需要保留。 评论区留言点赞: @浪里白条 帮大家总结了复习要点;@约书亚 的问题里提到了 MRR 优化;

索引 cb 的组织是先按 c 排序 , 在按 b 排序 , 同时记录主键

-c--|-b--|-主键部分a-- (同上)

上一篇 05 | 深入浅出索引(下) 下一篇 07 | 行锁功过:怎么减少行锁对性能的量

echo_陈 cho mysql 5.6不是支持online ddl了吗?也就是对表提作增加字段等功能,实际上不会阻塞读写? 4. 升级成MDL写锁 5. 释放MDL锁 4. 5如果没有锁冲突 执行时间非常短。第3步占用了DDL绝大部分时间,这期间这个表 可以正常读写数据,是因此称为 我们文中的例子,是在第一步就堵住了

np + -single-transaction 也是保证事务的一致性,但他只针对 有支持事务 引擎,比如 i nnodb 所以还是强烈建议大家在创建实例,表时候需要innodb 引擎 为好 全库只读 readonly = true 还有个情况在 slave 上 如果用户有超级权限的话 readonly 是失效的

当你做 dml 时候增加的 MDL 读锁, update table set id=Y where id=X; 并且由于隔离级别的 原因 法锁之间不油垒 加微信件82316662次筹更多

222 312

林晓斌 网络丁奇

Ilonetes FTWRL 納有读写的话。FTWRL 都会等待 读写执行宪毕后才执行 FTWRL 执行的时候要用触页的数据到磁盘 因为要保持数据的一 是 所有事务 都提交完毕的时候

表级别 锁 :一个直接就是表锁 lock table 建议不要使用, 影响太大 , 另个就是 MDL 元数据锁 MDL 是并发情况下维护数据的一致性。在表上有事务的时候,不可以对元数据经行写入操作,并且这 个是在server层面实现的