

# mysql MDL读写锁阻塞，以及online ddl造成的“插队”现象

原创

花落的速度

于 2019-07-18 14:27:26 发布


5104

收藏 51

版权

分类专栏：

mysql深入

mysql深入 专栏收录该内容

4 订阅

4 篇文章

订阅专栏

看mysql45讲遇到一个问题

session A	session B	session C	session D
begin;			
select * from t limit 1;			
	select * from t limit 1;		
		alter table t add f int; (blocked)	
			select * from t limit 1; (blocked)

我们可以看到 session A 先启动，这时候会对表 t 加一个 MDL 读锁。由于 session B 需要的也是 MDL 读锁，因此可以正常执行。

之后 session C 会被 blocked，是因为 session A 的 MDL 读锁还没有释放，而 session C 需要 MDL 写锁，因此只能被阻塞。

如果只有 session C 自己被阻塞还没什么关系，但是之后所有要在表 t 上新申请 MDL 读锁的请求也会被 session C 阻塞。前面我们说了，所有对表的增删改查操作都需要先申请 MDL 读锁，就都被锁住，等于这个表现在完全不可读写了。

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 再请求的话，这个库的线程很快就会爆满。

你现在应该知道了，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放，而会等到整个事务提交后再释放。

<https://blog.csdn.net/q2878948>

为什么C等待拿锁之后，D也会阻塞？其实这里并没有解释清楚。因为如果按并发理解的话，C，D应当是同等级，都有可能拿到锁的。但C读写锁互斥，D读读不互斥，这样的话就跟上图所述相悖了。就，查了一下。

(鸣谢 一梦如是YFL提供的文章)

首先是MDL（metaData Lock）的概念。元数据锁是server层的锁，表级锁，主要用于隔离DML（Data Manipulation Language，数据操纵语言，如select）和DDL（Data Definition Language，数据定义语言，如改表头新增一列）操作之间的干扰。每执行一条DML、DDL语句时都会申请MDL锁，DML操作需要MDL读锁，DDL操作需要MDL写锁（MDL加锁过程是系统自动控制，无法直接干预，读读共享，读写互斥，写写互斥）

花落的速度

关注

37







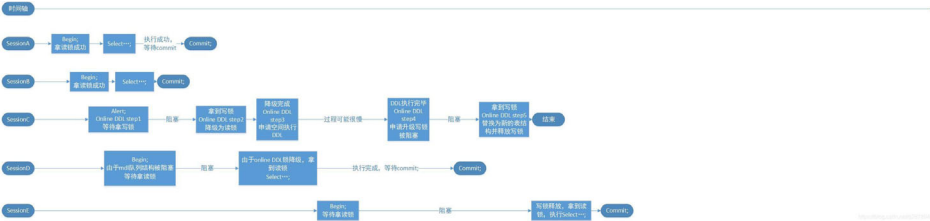
由于ddl执行时如果锁表的话会严重影响性能, 不锁表又难搞定操作期间dml语句的影响, 于是mysql推出了全新的online ddl概念, 即通过:

1. 拿MDL写锁
2. 降级成MDL读锁
3. 真正做DDL
4. 升级成MDL写锁
5. 释放MDL锁

5个步骤, 第一步拿读锁是为确保没有其他ddl语句在执行; 第三步是自己申请一块空间开始改表结构、填数据; 等填好了之后, 执行第四步, 这期间由于持有读锁, 可以确保不会有其他ddl语句造成不一致性; 最后等拿到写锁, 把表一替代就搞定了。

这样就看出来端倪了。上图中session A, Bcommit后, sessionC确实拿到了写锁, 只不过由于锁降级, 令sessionD拿到了读锁。但session D没有commit, 因此session C在执行online commit到第三步后, 又给阻塞了。所以就出现了类似于“插队”的现象。

如果想验证的话, 可以再开一个begin;select...的session, 就叫sessionE吧。然后让刚刚没commit的session D commit一下, 会发现这次session C并没有阻塞



MDL文件读取器

10-13

用于读取半条命mdl模型,对爱好OpenGL游戏编程的有一定的用处,可以把cs中得模型读出

详细分析mysql MDL元数据锁

09-08

主要介绍了mysql MDL元数据锁的相关资料, 文中讲解非常细致, 代码帮助大家更好的理解和学习, 感兴趣的朋...

评论 20

请发表有价值的评论, 博客评论欢迎灌水, 良好的社区氛围需大家一起维护。

评论

菠萝、 2022.01.26

按第一张图来看, 执行完四个步骤后, C 降级到读锁, D 也为读锁。  
此时如果 C 用 SELECT 一下, 而 D 使用 DDL 语句, D 上升到写锁但 D 不会被阻塞。  
经测试原因是: C 之前 alter 修改表结构时, 跟博主所述一样, 会自动提交当前事务, 并且不会再 begin 开启新的事务(当然, alter本身自己就是一个事务)。也就是说修改表结构后, 事务已经终止了。  
所以 C 即便 SELECT, 它也随着 SELECT 开始时本身这条语句事务的创建, 获得读锁, 到查询完结束这条语句自动提交并释放读锁, 所以 D 直接就修改了表结构

菠萝、 回复 菠萝、 2022.01.26

比如可以测试一下: RR 级别下, 此时 C 用 alter 之后, 使用 update 后, 另一个客户端在不开启事务时 select 是可见的, 并没有被隔离。说明 C 已经不处于事务之中了

键盘上的艺术家w 2021.08.10

1

大佬 怎么把 极客时间的课程搞成pdf啊?

little鸿鸿 2021.05.26

第一遍看没啥感觉, 看第二遍真的学习了, 还实验了一把, 加了session e, 果然查询的时候, 查询出了 session c加的字段, 666

mysql数据库锁MDL锁的解释\_huan的学习记录

2-17

MDL不需要显示使用,在进行表操作时会自动加上。当对表进行增删改查时,会自动加上MDL读锁;当要对表进行加...

mysql mdl 锁\_MySQL MDL锁\_AI Starter的博客

session A 通过 lock table 命令持有表 t 的 MDL 写锁,而

花落的速度

关注