

点赞再看，养成习惯，微信搜索【三太子敖丙】关注这个互联网苟且偷生的工具人。

本文 **GitHub** <https://github.com/JavaFamily> 已收录，有一线大厂面试完整考点、资料以及我的系列文章。

前言

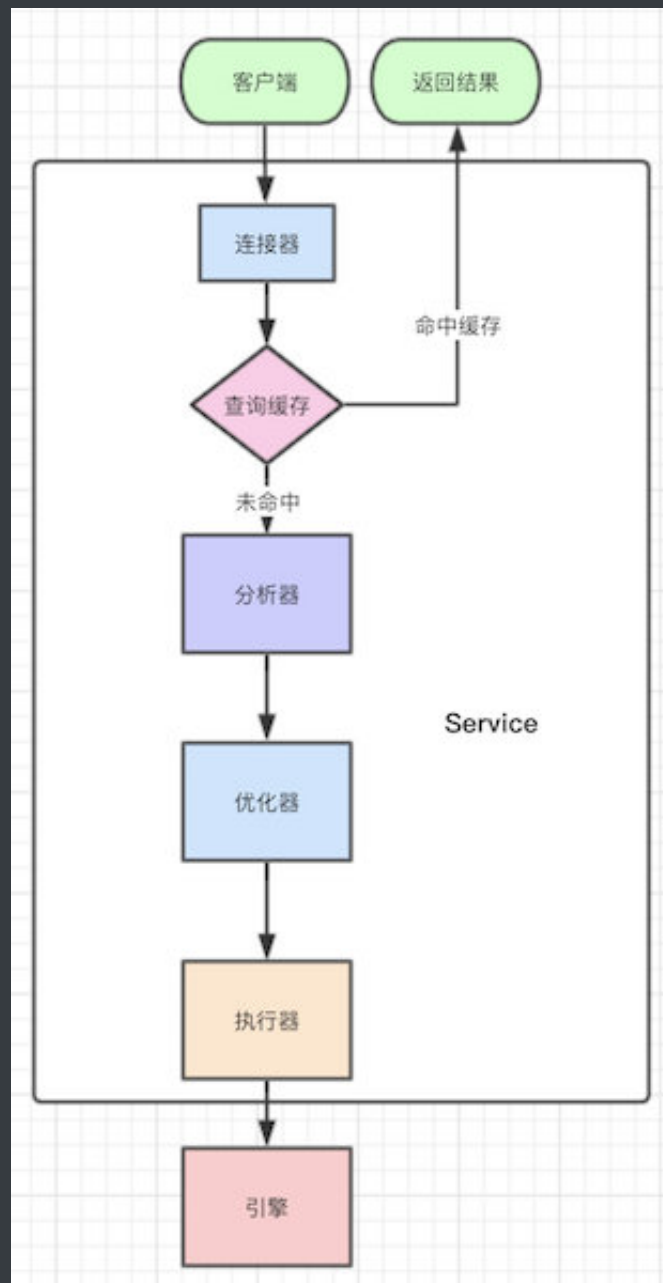
我是个标题党，所有文章的名字只是我的噱头，我们应该有一颗谦逊的心，所以希望大家怀着空杯心态好好学，一起进步。

数据库我想大家应该一点都不陌生吧，我想不管你写啥的，数据库就算没用过也听过了，是我们项目体系里面不可或缺的一环。

问了一圈，身边朋友公司要么用自研，要么就清一色的MySQL，所以想看Oracle的朋友可能要失望了，但是不影响你了解数据库的通用知识。

正文

你知道MySQL的基本架构么？你能在纸上给我大致画出这个示意图么？



好的那我们按照顺序了解下，连接器是啥？

我们要进行查询，第一步就是先去链接数据库，那这个时候就是连接器跟我们对接。

他负责跟客户端建立链接、获取权限、维持和管理连接。

链接的时候会经过TCP握手，然后身份验证，然后我们输入用户名密码就好了。

验证ok后，我们就连上了这个MySQL服务了，但是这个时候我们处于空闲状态。

怎么查看空闲连接列表？

`show processlist`, 下图就是我在自己的数据库表执行命令的结果, 其中的Command列显示为**Sleep**的这一行, 就表示现在系统里面有一个空闲连接。

Id	User	Host	db	Command	Time	State	Info
13627990	test	10.70.1.60:605	mc	arch Sleep	1560		(NULL)
13628490	test	10.70.1.10:8	mo	arch Sleep	780		(NULL)
13628547	test	10.70.1.10:8	mo	arch Sleep	690		(NULL)

这里需要注意的是, 我们数据库的客户端太久没响应, 连接器就会自动断开了, 这个时间参数是**wait_timeout**控制住的, 默认时长为8小时。

断开后重连的时候会报错, 如果你想再继续操作, 你就需要重连了。

这个有个我看过的书本的案例:

一个在政府里的朋友说, 他们的系统很奇怪, 每天早上都得重启一下应用程序, 否则就提示连接数据库失败, 他们都不知道该怎么办。

我分析说, 按照这个错误提示, 应该就是连接时间过长了, 断开了连接。数据库默认的超时时间是8小时, 而你们平时六点下班, 下班之后系统就没有人用了, 等到第二天早上九点甚至十点才上班, 这中间的时间已经超过10个小时了, 数据库的连接肯定就会断开了。

是的, 就是超出了超时时间, 然后写代码的人也没注意到这个细节, 所以才会出现这个问题。

把超时时间改得长一点, 问题就解决了。

这种参数其实我们平时不一定能接触到, 但是真的遇到问题的时候, 知道每个参数的大概用法, 不至于让你变成无头苍蝇。

那除了重新链接, 还有别的方式么? 因为建立链接还是比较麻烦的。

使用长连接。

但是这里有个缺点, 使用长连接之后, 内存会飙得很快, 我们知道MySQL在执行过程中临时使用的内存是管理在连接对象里面的。

只有在链接断开的时候才能得到释放, 那如果一直使用长连接, 那就会导致OOM (Out Of Memory), 会导致MySQL重启, 在JVM里面就会导致频繁的Full GC。

那你会怎么解决?

我一般会定期断开长连接，使用一段时间后，或者程序里面判断执行过一个占用内存比较大的查询后就断开连接，需要的时候重连就好了。

还有别的方法么？你这种感觉不优雅呀小老弟。

执行比较大的一个查询后，执行`mysql_reset_connection`可以重新初始化连接资源。这个过程相比上面一种会好点，不需要重连，但是会初始化连接的状态。

你了解MySQL的查询缓存么？

MySQL拿到一个查询请求后，会先到查询缓存看看，之前是不是执行过这条语句。

大家是不是好奇同一条语句在MySQL执行两次，第一次和后面的时间是不一样的，后者明显快一些，这就是因为缓存的存在。

他跟Redis一样，只要是你之前执行过的语句，都会在内存里面用key-value形式存储着。

查询的时候就会拿着语句先去缓存中查询，如果能够命中就返回缓存的value，如果不命中就执行后面的阶段。

但是我还是不喜欢用缓存，因为缓存弊大于利。

哦？此话怎讲？

缓存的失效很容易，只要对表有任何的更新，这个表的所有查询缓存就会全部被清空，就会出现缓存还没使用，就直接被清空了，或者积累了很多缓存准备用来着，但是一个更新打回原形。

这就导致查询的命中率低的可怕，只有那种只查询不更新的表适用缓存，但是这样的表往往很少存在，一般都是什么配置表之类的。

那我们查询的时候不想用缓存一般都是怎么操作的，或者是用缓存又怎么操作？

可以显示调用，把`query_cache_type`设置成为DEMAND，这样SQL默认不适用缓存，想用缓存就用SQL_CACHE。

有个小技巧就是，我们之前开发的时候，都会去库里看看sql执行时间，但是可能是有缓存的，一般我们就在sql前面使用SQL_NO_CACHE就可以知道真正的查询时间了。

```
select SQL_NO_CACHE * from B
```

缓存在MySQL8.0之后就**取消**了，所以大家现在应该不需要太关注这个问题，主要是我之前用的版本都不高，所以缓存一直有，在《高性能MySQL》书中也看到了一些关于缓存的介绍，就想起来给大家也提一下了。

缓存查询完了应该做啥呢？

在缓存没有命中的情况下，就开始执行语句了，你写的语句有没有语法错误，这是接下来MySQL比较关心的点。

那他会怎么做呢？会先做**词法分析**，你的语句有这么多单词、空格，MySQL就需要识别每个字符串所代表的是什么，是关键字，还是表名，还是列名等等。

然后就开始**语法分析**，根据词法分析的结果，语法分析会判断你sql的对错，错了会提醒你的，并且会提示你哪里错了。

message

1064 - You have an error in your SQL syntax; check the manu

分析没错之后就进入下一步，**优化器**。

主要是优化什么呢？

优化就比较简单了，因为我们建立表可能会建立很多索引，优化有一步就是要确认使用哪个索引，比如使用你的主键索引，联合索引还是什么索引更好。

还有就是对执行顺序进行优化，条件那么多，先查哪个表，还是先关联，会出现很多方案，最后由优化器决定选用哪种方案。

最后就是执行了，执行就交给执行器去做。

第一步可能就是权限的判断，其实这里我不确定的一个点就是，我接触的公司很多都是自研的线上查询系统，我们是不能用Navicat直连线上库，只能去网页操作，那表的权限是在MySQL层做的，还是系统做的，我猜应该是系统层做的，MySQL可能默认就全开放了，只是我们不知道ip。

有知道的小伙伴也可以在公众号【三太子敖丙】去加我微信跟我说。

执行的时候，就一行一行的去判断是否满足条件，有索引的执行起来可能就好点，一行行的判断就像是接口都提前在引擎定义好了，所以他比较快。

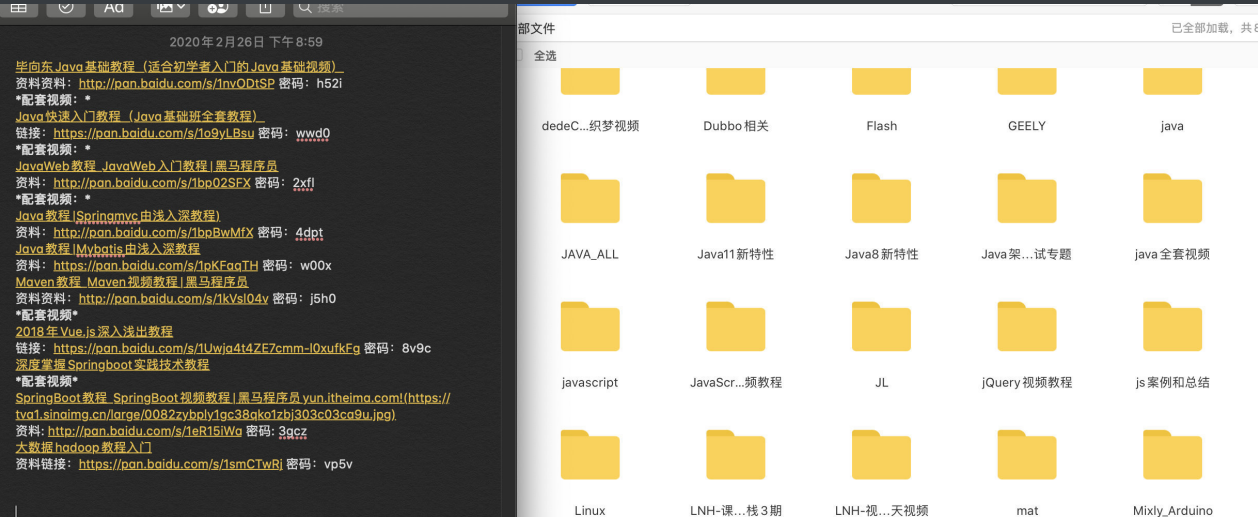
数据库的慢日志有个**rows_examined**字段，扫描多少行可以看到，还有**explain**也可以看到执行计划，我们扫描了多少行。

可以小伙子，基础大致框架还是了解得很清楚的，我们下次深入了解下，索引和部分机制。

好的，我们下次见。

相关资料

Tip：本来这一栏有很多我准备的资料的，但是都是外链，或者不合适的分享方式，博客的运营小姐姐提醒了我，所以大家去公众号回复【资料】好了。



总结

基本上我把MySQL的逻辑架构的东西都简单聊了一遍，当然你去自信了解的话，你会发现其实里面还有很多细节的，我只是说了一些常见的问题，这还是阿里丁奇学长的《MySQL实战》的思路。

我自己在MySQL方面更多的可能就是理论知识了，还做不到深入了解的地步，大家如果有机会一定要深入的去学习一下。

絮叨

这应该是我开年的第一篇技术文，本来年前写了一点索引的东西，但是后面在构思的时候想了想，还是从数据库的基础知识跟大家熟悉起来，再去了解索引会好点。

新年可能会尝试一些新的文章风格，不一定全部用面试的方式，当然这篇是面试的风格，因为新的风格还没决定。我还会尝试着在4月份以后开始以视频录制这样的方式尝试几期，大家有什么意见也可以在微信上跟我说。

以后絮叨我也放最后，这样大家阅读体验好点，反正有啥直接跟我反馈，宠粉的我，纷纷安排。

大家最近一定穿好衣服不要感冒啥的，鸡蛋去看个病，差点吓尿了，给你们看看他的装备：



我是敖丙，一个在互联网苟且偷生的工具人。

创作不易，不想被白嫖，各位的「三连」就是丙丙创作的最大动力，我们下次见！

文章持续更新，可以微信搜索「三太子敖丙」第一时间阅读，回复【资料】【面试】【简历】有我准备的一线大厂面试资料和简历模板，本文 **GitHub** <https://github.com/JavaFamily> 已经收录，有大厂面试完整考点，欢迎Star。



你知道的越多，你不知道的越多