

腾讯云DBA岗位的面试题解析

来自： 小6互联网求职面试

 马听

2023年11月19日 11:02



扫码加入面试交流群
查看更多优质面试题

前几天分享了腾讯云DBA岗位的面试题，今天来解析一下这些面试题。
当然，可能解析的不全面，欢迎在留言区补充
另外一般还是建议在面试回答的时候，结合自己的工作经历，比如MySQL监控，能讲出自己公司独有的，能让面试官眼前一亮，可能就是加分项。

1 这几年从事哪些和 MySQL 有关的工作

这个根据自己情况讲

2 常见监控项

系统相关

- 系统cpu_iowait
- 系统cpu负载
- 系统内存交换
- 系统文件描述符
- 系统网络流量（进出）
- 磁盘利用率

MySQL 状态相关

- MySQL 进程状态
- QPS
- TPS
- 慢查询数量

锁相关

- 表锁
- 行锁
- 死锁
- 锁等待

连接相关

- 最大连接数
- 活跃连接

主从

- io sql线程
- 从库read_only
- 主从延迟
- 主从切换

业务相关

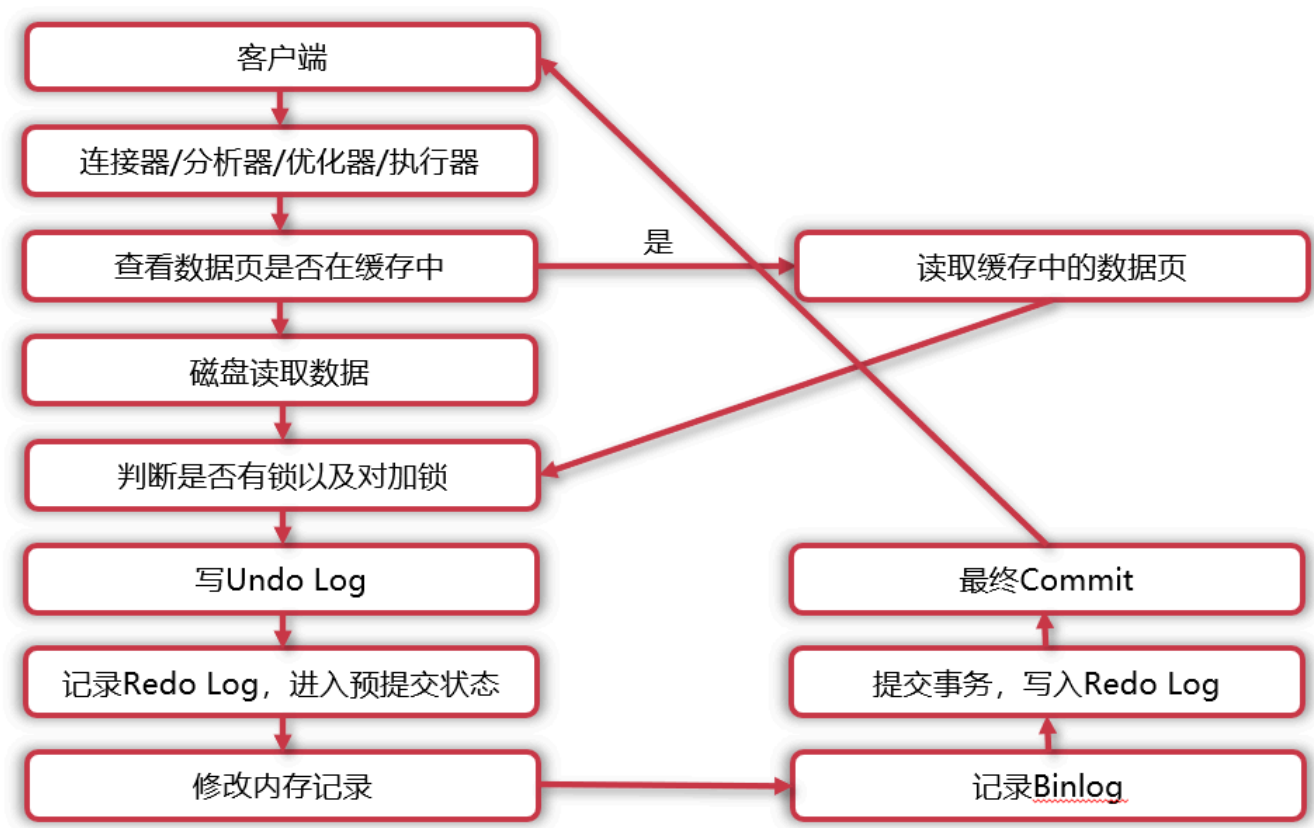
- 主键自增值
- 表数据量

3 自增主键到上限了会发生什么

表的自增 id 达到上限后，再申请时它的值就不会改变，进而导致继续插入数据时报主键冲突的错误。
row_id 达到上限后，则会归 0 再重新递增，如果出现相同的 row_id，后写的数据会覆盖之前的数据。

4 一条 update 会经历哪些过程

一条更新语句的生命历程



首先是连上连接器，负责跟客户端建立连接、获取权限、维持和管理连接

分析器会先做词法分析。需要确定SQL语句中的字符串分表是什么，代表什么？

再做语法分析，判断输入的SQL是否满足MySQL语法。

再经过优化器，决定是否走索引，走哪个索引，或者决定关联表的顺序

执行器负责具体的执行，

进入到存储引擎层，

首先查看数据页是否在缓存中，如果在，就读取数据，如果不在，就从磁盘中读取

判断要操作的记录是否存在锁，如果存在，就等待

如果不存在，就对要修改的记录行加锁

再写Undo Log

记录Redo Log，进入预提交状态

修改内存中的记录

记录Binlog

提交事务，写入Redo Log，

最终Commit

5 怎样设计一个高可用高性能的 MySQL 集群

可以从两个方向考虑：

第一种，类似Orch的方案，也就是借助第三方工具去探测主库，当然，第三方工具可能有至少3台机器，当3台机器有两台认为主库宕机，则进行切换

第二种，类似MGR的，就是节点之间互相探测，如果多数节点认为主库宕机了，则切换

而切换又涉及到选主，可以根据优先级，以及从库和主库的延迟，优先级高的，以及延迟底的，优先考虑作为新主

6 MySQL CPU 高负载怎么排查

查看进程列表：用top查看哪些进程占用了大量CPU资源

检查MySQL连接：在MySQL中，执行show processlist，查看是否有正在执行的很占CPU资源的SQL，以前就遇到过，研发执行全表delete，导致CPU跑慢的情况

检查慢查询日志：查看慢查询日志，看从CPU高负载开始到现在，所有的慢查询，可以确定是不是慢查询导致的，如果有慢查询，就需要考虑是否要添加索引，是否要优化SQL等

观察其他指标：比如是否有锁等待，QPS和TPS是否有暴涨的情况等。

7 怎么分析慢查询

explain：查看SQL执行类型，扫描行数和是否走索引等；
show profile：确定 SQL 执行过程具体在哪个过程耗时比较久；
trace：查看优化器如何选择执行计划。

8 pt 工具分析慢查询有哪些作用

可以获取慢查询统计信息，比如某个时间段，哪几类SQL最耗时间
另外可以把慢查询分析结果存放到MySQL表中，然后通过Grafana或者开发一个前端页面进行展示

9 事务繁忙的 MySQL，物理机宕机，启动后会发生什么？具体怎么判断哪些要回滚，哪些要提交

在事务执行过程中，首先会将操作写入redo log，并刷新到磁盘。宕机恢复后，MySQL会通过检查redo log来确定哪些事务已经完成但尚未提交，这些事务将会被提交。

当MySQL发现某个事务在宕机时尚未完成，它会利用undo log来回滚这个事务。

10 怎么判断主从延迟

Seconds_Behind_Master

一种常规的方法就是 show slave status 查看 Seconds_Behind_Master，这个参数表示从库延迟的秒数。
如果是0，表示可能没有延迟。这里为什么是可能呢？

当从库正在主动处理更新时，此字段显示从库上的当前时间戳与从库上当前正在处理的事件的主库上记录的原始时间戳之间的差异。

当副本上当前没有处理任何事件时，该值为 0

在某些情况下，Seconds_Behind_Master 并不一定准确。比如网络中断时，Seconds_Behind_Master = 0，并不能代表主从无延迟。

因此，有比这个更准确的一种方法：对比位点或 GTID。

对比位点

如果是基于位点的复制，则判断 Master_Log_File 跟 Relay_Master_Log_File 是否相等，如果 Relay_Master_Log_File 落后 Master_Log_File，则表示主从存在延迟。

其中

Master_Log_File 表示 IO 线程正在读取的主库 binlog 文件名

Relay_Master_Log_File 表示SQL 线程最近执行的事务对应的主库 binlog 文件名

或者判断 Read_Master_Log_Pos 跟 Exec_Master_Log_Pos 是否相等，如果后者落后前者很多，则表示延迟比较高。

其中

Read_Master_Log_Pos 表示IO 线程正在读取的主库 binlog 文件中的位点

Exec_Master_Log_Pos 表示 SQL 线程最近读取和执行的事务对应的主库 binlog 文件中的位点

对比GTID

如果开启了 GTID 复制，则可以对比 Retrieved_Gtid_Set 和 Executed_Gtid_Set 是否相等，如果 Executed_Gtid_Set 落后很多，则表示存在延迟。

其中

Retrieved_Gtid_Set：从库收到的所有日志的 GTID 集合；

Executed_Gtid_Set：从库已经执行完的 GTID 集合。

11 performance_schema 和 information_schema 的区别

保存信息的差异

performance_schema：包含资源消耗、资源等待等情况

information_schema：主要关注 server 运行过程中的元数据信息

使用场景的差异

performance_schema主要用于性能调优、故障排查和性能监控，可用于诊断数据库性能瓶颈、分析查询执行情况等。

information_schema用于查询和了解数据库对象的结构、权限、状态等信息。

12 一个实例活跃线程个数多少以下算合理？

注意，这里是活跃线程个数，不是连接数，活跃线程个数一般建议是建议为机器核数的两倍值以下；

因为活跃线程往往才会消耗CPU；

通常也建议设置最大的活跃线程个数，MySQL可以用innodb_thread_concurrency参数来控制最大的活跃线程个数。