

12月3日模拟面试解析

来自： 小6互联网求职面试

 马听

2023年12月04日 21:22



扫码加入面试交流群
查看更多优质面试资料

在12月3日，有用户请嘉宾@荣屹 老师帮忙进行了一场模拟面试。

之前只发了题目， 这里来分享一下解析：

1 讲下MySQL二阶段提交， Prepare阶段， 机器宕机怎么办， commit阶段， 机器宕机恢复， 可能做哪些操作？
两阶段提交流程：

- 1 Redo log的Prepare
- 2 写Binlog
- 3 Redo Log 的commit

Prepare阶段， 机器宕机重启之后， 因为Binlog还没写， Redo log也没提交， 那MySQL启动之后， 事务回滚就行
commit阶段， 机器宕机重启之后，
就需要做一些判断
如果Redo log是完整的， 则事务直接提交
如果Redo log里的事务只是Prepare状态， 就需要判断Binlog是否完整：
Binlog完整， 就提交事务， Binlog不完整， 就回滚事务。

2 MySQL根据锁粒度， 可以分哪几种锁类型

- 全局锁
- 表级锁
- 行锁

3 为什么会出现死锁， 列出一种可能出现死锁的场景

两个或者多个事务在同一资源上相互占用， 并请求锁定对方占用的资源， 从而导致恶性循环的现象。
session1更新了某张表a等于1的这行， 没提交
session2更新了这张表a等于2的这行， 也没提交
session1再更新了a等于2的这行， 会等待
session2再更新了a等于1的这行， 就会产生死锁

4 数据库层面遇到死锁会怎么处理

通常MySQL会回滚一个事务。

5 讲一下MDL锁， 以及经常可能出现的场景

MDL： 当对一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。
可能出现MDL锁等待的场景： 表结构变更， 表数据备份和恢复时。

6 MySQL Redis、MongoDB常见监控项

Redis常见监控项

- 连接失败监控
- 客户端连接数
- 配置的最大内存
- 最大内存策略
- 角色监控
- 复制状态监控
- 延迟监控
- 从库是否设置只读
- QPS
- 网络总出/入量
- 每秒输出/入量
- 内存使用率
- 内存碎片率
- 缓存命中率
- key数量
- 大key

热key
慢查询

MongoDB常见监控项

连接
内存
锁
命令执行统计
副本集状态

7 MySQL所在的机器，CPU负载高，怎么定位和分析？
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。

8 MySQL慢日志，里面有哪些信息？

- Query_time：语句执行时间，单位为秒。
- Lock_time：获取锁的时间，单位为秒。
- Rows_sent：发送给客户端的行数。
- Rows_examined：MySQL Server层检查的行数。

从MySQL 8.0.14开始，新增了log_slow_extra参数，开启之后，可以查看一些额外信息，比如：

- Thread_id：语句线程ID。
- Errno：语句错误号。
- Killed：如果该语句终止，则错误号指示原因。如果该语句正常终止，则返回0。
- Start：语句开始执行的时间。
- End：语句结束执行的时间

9 MySQL怎么定位最耗时或者最消耗性能的SQL

一般就是通过慢查询记录中，扫描行数最多的，很可能就是最消耗性能的
当然，用pt-query-digest分析的话，一般是排行前几的SQL，就是最消耗性能的。

10 假如研发执行了一条update，忘记加条件了，怎么恢复？
可以通过一些工具，解析出Binlog中误操作update语句的回滚语句，比如my2sql工具。

11 MySQL 5.6，出现主从延迟，怎么处理？

如果是多个库的并发高，可以开启库级别的多线程复制
如果是单个库，可以调整从库的innodb_flush_log_at_trx_commit和sync_binlog，暂时不要设置为双一。

12 执行计划中，type=all表示什么？

type=index，Extra是using where表示什么？

type=all 表示全表扫描

type=index，Extra是using where表示

可能是索引+其他字段的条件查询，也有可能是回表的查询

13 explain中filtered这个字段表示什么？

按条件筛选的行的百分比

14 执行计划中，出现：Index Condition Pushdown，表示什么？有什么好处？

表示ICP，在没有ICP的情况下，存储引擎遍历索引以定位表中匹配的行数据，并将这些行数据返回给 MySQL服务，MySQL服务对这些行再次进行where条件的过滤。启用ICP后，在取出索引的同时，MySQL服务将where条件下推到存储引擎。存储引擎使用索引项来评估推入的索引条件，只有满足这个条件，才从表中读取行。启用ICP可以减少存储引擎访问表的次数和MySQL服务访问存储引擎的次数。

15 深分页limit怎么优化

比如：limit 100000,10;

主键连续，就直接通过主键过滤数据

先查主键，再根据主键匹配数据

16 MongoDB读写分离怎么做的

ReadPreference

primary：主节点，默认模式，读操作只在主节点，如果主节点不可用，报错或者抛出异常。

primaryPreferred：首选主节点，大多情况下读操作在主节点，如果主节点不可用，如故障转移，读操作在从节点。

secondary：从节点，读操作只在从节点，如果从节点不可用，报错或者抛出异常。

secondaryPreferred：首选从节点，大多情况下读操作在从节点，特殊情况（如单主节点架构）读操作在主节点。

nearest：最邻近节点，读操作在最邻近的成员，可能是主节点或者从节点。

17 MongoDB有哪些高可用方案？分别在何种场景使用

副本集和分片

一般副本集够用，如果数据量过大，副本集实在满足不了我们的需求，才建议去考虑使用分片集群

18 MongoDB备份工具，mongodump和mongoexport区别是什么？

mongodump导出的是bson格式，不能直接查看

mongoexport导出的是json或者csv。

19 Redis 怎么找到大key或者热key

大key发现：

redis-cli --bigkeys，可以看到每种类型的最大key

通过程序查询Bigkey

通过rdbtools来找到Bigkey（最推荐）

热key发现：

redis-faina

redis-cli --hotkeys（最大内存策略需要修改成volatile-lfu 或 allkeys-lfu，他的逻辑就是优先淘汰访问最少的key

所以他就会记录每个key的访问频率。就能方便我们通过hotkeys来获取到热key)

20 Redis怎么安全的删除大key

string类型：4.0之后，使用unlink 接key名，可以惰性删除，另外从Redis6.0开始，开启lazyfree-lazy-user-del参数

则普通的del，跟unlink效果一样，采用惰性删除

hash或者list类型，每次删一部分元素，循环删除。

21 使用Redis Cluster遇到过什么问题吗？

看自己的情况讲，

比如Redis Cluster节点过多，可能客户端获取槽的元数据都会很慢

22 MySQL高可用方案有哪些？

Orchestrator

InnoDB Cluster

双主+Keepalived