

## 1.碰到线上问题，你的解决思路是怎样的？

在遇到线上问题之后，首要目标是迅速恢复服务或减少对服务的影响。总体的思路可以按照故障发现--故障降级--故障解决--故障避免四个步骤来进行。一般通过监控或者是业务人员上报反馈发现问题，进一步可以通过做功能降级，比如关闭非核心功能等。然后结合一些流量降级手段比如限流熔断来降低故障的影响范围。第三部就可以通过一些常用的手段定位故障完成故障修复，在修复完验证之后上线。最后对事故进行复盘总结，做好事故报告，防止同类事故再次发生



## 2.线上出现Mysql死锁，如何排查

首先要明确发生Mysql死锁后，程序会出现什么现象，如果Mysql未开启死锁检测，发生死锁程序通常会出现卡死现象，如果开启了死锁检测，Mysql通常会自动回滚一个事务，结束死锁。明确了死锁现象后，就需要通过SHOW ENGINE INNODB STATUS来查看innodb的状态信息，从中寻找"LATEST DETECTED DEADLOCK"或类似的关键字然后根据以上关键字下显示的详细信息分析死锁原因。然后根据具体的死锁原因完成对应的修复，修复完之后上线负载测试，最后一般会执行SET GLOBAL innodb print all deadlocks=1命令打开死锁监控，这样Mysql就会把所有死锁信息记录到MySQL错误日志中，方便后续定位。同样也可以使用性能监控工具如Percona Toolkit、MySQL Enterprise Monitor等)实时监控数据库的性能指标

-- 长期开启死锁日志记录

```
SET GLOBAL innodb_print_all_deadlocks=1;
```



- innodb print all deadlocks默认=0，仅记录最近一次发生的死锁信息（会被后续死锁信息覆盖），这个信息不会自动写入错误日志文件，而是保存在内存中，必须通过 SHOW ENGINE INNODB STATUS 命令才能查看到
- 错误日志是error.log

### 3.线上出现数据库连接池爆满如何排查？

首先确定问题现象，由于现在的应用一般都会采用orm框架来开发，当应用出现数据库连接爆满情况的时候，框架会打印出对应日志，帮助我们确定问题。在出现连接池耗尽的时候，Go程序里的gorm框架会报 Too many connections 错误，我们可以通过类似的标志性的报错信息确定问题是由数据库连接引起的。接下来在mysql端可以使用SHOW PROCESSLIST命令来查看各个连接的使用情况，这里可看到具体在使用的连接数，以及多少连接正被占用没有释放，通过连接数我们可以确定是否是最大连接数配置过小，可以优化连接数配置。然后定位到具体的sql，是哪个sql占用着连接没有释放，从而到代码中去优化对应的sql或代码，根据具体问题做具体优化。优化完问题之后，再上线作负载测试同时做好数据库的连接使用监控



- SHOW PROCESSLIST 是 MySQL 中用于查看当前所有连接线程状态的重要命令，它显示服务器中正在运行的所有客户端连接和查询信息

### 4.Go程序里的pprof性能分析工具有用过吗，简单介绍下怎么用pprof工具定位优化Go程序性能？

top->list 函数名->web（需要安装graphviz）

pprof工具可以十分方便的定位cpu，内存，锁竞争，协程泄漏等常见的程序性能问题，在使用上首先需要import pprof工具包，这样程序就会每隔一段时间自动采集当前程序的堆栈信息，随后我们就可以排查具体指标了。在终端可以直接执行 go tool pprof 命令来分析具体指标信息，然后通过 top，list 命令来查看具体的异常指标，就可以明确定位到问题代码的位置。同时我们还可以安装graphviz工具，将上述生成的堆栈信息文件用图形化的形式在浏览器端打开，用图形可视化的方式查看各个指标的情况，一般存在异常的话，对应的指标框会显示的越大，这样我们可以直接定位到代码位置，完成修改

指标	描述	备注
allocs	内存分配信息	可以用浏览器直接点击打开
blocks	阻塞操作信息	可以用浏览器直接点击打开
cmdline	显示程序启动命令及参数	点击浏览器打开会下载文件
goroutine	当前所有协程的堆栈信息	可以用浏览器直接点击打开
heap	堆上内存使用信息	可以用浏览器直接点击打开
mutex	锁竞争信息	可以用浏览器直接点击打开
profile	CPU 占用信息	点击浏览器打开会下载文件
threadcreate	系统线程创建信息	可以用浏览器直接点击打开
trace	程序运行跟踪信息	点击浏览器打开会下载文件

阻塞操作排查: block

协程泄漏排查: goroutine

内存占用过多排查: heap

频繁GC排查: heap+

```
GODEBUG=gctrace=1 ./pprof-demogrep gc
```

锁竞争排查: mutex

cpu占用过高排查: profile