**性能测试流程及常见问题总结**

**一、产品需求**

**1、业务场景：**

一个问卷调查的功能，然后产品和业务会不定时通过前端界面去根据筛选条件查询相关问卷问题的答案明细，但是觉得很慢，让测试这边给出一个指标。

**2、系统架构：**

MySQL数据库，所有问卷问题相关的数据都存储在同一张表，单台服务器，无缓存，通过一个查询接口去查询返回数据。

**3、数据量：**

每天大概新增3000张问卷调查，每张问卷30个问题，每个问题下面还有具体的答案，答案的数据类型、大小不清楚。

PS：从我个人的了解来看，对大部分测试人员来说，遇到的性能需求大体都是这种范围不明确，指标不清楚的性能需求，那么如何做好测试工作，在体现自己价值的同时，还能学习提升呢？

**二、具体问题具体分析**

**1、场景建模**

和产品业务沟通，明确他们的操作场景，比如查询的条件（时间范围、问卷类型、分数范围、用户类型等），操作时间（具体到每天哪个时间段有多少人进行这些操作）。

**2、确定指标**

明确了业务场景后，确认不同的操作下，用户（这里是产品和业务人员）的可接受值（比如每天早晨9:00-9:10，100个人进行查询操作，查询条件是最近一周A类型用户的B类型问卷，分数在80分以上），

可接受的最大响应时间不超过5S。

**三、测试进行中**

确认测试范围和具体的性能指标后，接下来就需要进行测试方案设计、测试用例设计等一系列的计划了，这个阶段是最耗费时间也是最麻烦的。

**1、环境确认**

首先需要确认测试的执行环境，是生产、UAT还是独立的测试环境。测试环境对测试结果的影响是很大的，大体如下：

**生产：**在执行测试的过程不能对其他用户访问造成影响（时间选择很重要），测试数据污染要解决（数据隔离：线程标记、用户白名单、挡板、mock对象、测试数据落入影子库）；

**UAT：**作为验收环境，一般来说数据的准确性和系统版本都是最新和相对稳定的，但要考虑对其他业务的影响，理由同上；

**测试：**数据预埋、基础数据准备、测试数据准备、每次执行迭代后的数据初始化、服务器配置和生产是否可以等量代换等；

**2、团队合作**

性能测试不是一个人就可以搞定的，一般都需要运维、DBA、开发、测试配合来进行，因此做好沟通和协作很重要。

**3、测试要做什么**

上面的工作做完之后，你需要考虑测试执行工具和脚本开发的工作。需要做的事情如下：

①、和开发沟通，获取业务功能对应的接口文档（如果没有，想办法），参数字段的含义，对应的数据库表字段，造成的影响；

②、和运维沟通，确确认服务器的部署，配置（这里可能需要进行基准测试和配置测试）；

③、和DBA沟通，确认测试数据预埋、基础数据准备、迭代后的数据初始化工作；

④、测试人员本身，需要准备测试数据，开发测试脚本，进行脚本调试，执行和监控分析等工作；

**四、体现测试的价值**

如何在性能测试中体现测试的价值？

我相信很多测试童鞋都经历过那种不被看中的阶段，但也要努力去改变现状，不断体现自己的价值。如何体现，请看下面：

**1、做好沟通协调**

①、和业务产品沟通，确认需求和场景；

②、和技术团队沟通，尽量多沟通，达成一致（测试方案、测试用例、测试数据、测试环境）；

**2、本职工作要做好**

①、测试方案、测试用例设计；

②、测试工具选型、测试脚本开发和调试；

③、测试数据的准备；

④、测试执行、监控和分析定位；

**3、创造价值才能赢得尊重**

职场，一切到头来还要从自身创造的价值来赢得尊重，那么如何从测试的角度创造价值？

①、提高交付的产品质量（覆盖率、风险分析、容错方案、容灾方案）；

②、提高交付速率（解决问题的过程抛出问题，流程不规范、开发不规范、管理不规范等，抛出问题，然后推动解决问题）

③、打铁也要自身硬！因此不断学习提高自己的技术能力，不断总结沟通，才能更好和同事交流，从解决问题的角度出发，去解决问题，创造价值！

**五、回归问题本身**

上面说的有点跑题了，回到问题本身，说说我对这个性能需求的一些优化建议吧，仅供参考：

**1、数据库优化**

问题点：从上面描述的情况来看，每天产生的数据大概有10W+条，且只有一张表存储；

解决方案：分库分表，表可以拆分为问卷主表、问卷对应的问题表、问题对应的答案明细表等，长期来说数据量不小，可以考虑分库，主从分离等，查询添加索引等方法。

**2、处理逻辑优化**

问题点：一次性查询的数据过多，导致前端展示较慢；

解决方案：查询结果分批次展示（比如有100W条数据，分为100个批次，每个批次10000条数据）。

**3、存储优化**

问题点：没有缓存，直接从DB单表读取，容易造成超时和表锁；

解决方案：将数据放入缓存服务器（比如Redis），设定查询次数或者有效时间，多级缓存，提高缓存命中，防止缓存穿透和同时失效带来的瞬间DB压力。

**4、业务优化**

问题点：多人短时间内查询大量数据，对服务造成巨大压力；

解决方案：和产品业务沟通，让查询操作时间在业务平缓期，拉长查询操作的时间线等。

**5、服务优化**

问题点：单台服务器；

解决方案：做服务集群和负载均衡，增加监控，设定阈值，超过阈值则临时增加新的服务器，分流。