**完整的软件性能测试流程及案例**

我们在进行性能测试工作的过程中，需要借助工具的辅助来帮我们完成一些工作，但loadrunner≠性能测试！或者说，性能测试工具≠性能测试，工具永远是一种辅助的工具，而不能认为会用工具就会性能测试了！下面，就说说一个完整的性能测试过程吧。。。

**PS：**文末附上一张性能测试的思维导图

**一、准备工作**

**1、系统基础功能验证**

性能测试在什么阶段适合实施？切入点很重要！一般而言，只有在系统基础功能测试验证完成、系统趋于稳定的情况下，才会进行性能测试，否则性能测试是无意义的。

**2、测试团队组建**

根据该项目的具体情况，组建一个几人的性能测试team，其中DBA是必不可少的，然后需要一至几名系统开发人员（对应前端、后台等），还有性能测试设计和分析人员、脚本开发

和执行人员；在正式开始工作之前，应该对脚本开发和执行人员进行一些培训，或者应该由具有相关经验的人员担任。

**3、工具的选择**

综合系统设计、工具成本、测试团队的技能来考虑，选择合适的测试工具，最起码应该满足一下几点：

①支持对web（这里以web系统为例）系统的性能测试，支持http和https协议；

②工具运行在Windows平台上；

③支持对webserver、前端、数据库的性能计数器进行监控；

**4、预先的业务场景分析**

为了对系统性能建立直观上的认识和分析，应对系统较重要和常用的业务场景模块进行分析，针对性的进行分析，以对接下来的测试计划设计进行准备。

**二、测试计划**

测试计划阶段最重要的是分析用户场景，确定系统性能目标。

**1、性能测试领域分析**

根据对项目背景，业务的了解，确定本次性能测试要解决的问题点；是测试系统能否满足实际运行时的需要，还是目前的系统在哪些方面制约系统性能的表现，或者，哪些系统因素导致

系统无法跟上业务发展？确定测试领域，然后具体问题具体分析。

**2、用户场景剖析和业务建模**

根据对系统业务、用户活跃时间、访问频率、场景交互等各方面的分析，整理一个业务场景表，当然其中最好对用户操作场景、步骤进行详细的描述，为测试脚本开发提供依据。

**3、确定性能目标**

前面已经确定了本次性能测试的应用领域，接下来就是针对具体的领域关注点，确定性能目标（指标）；其中需要和其他业务部门进行沟通协商，以及结合当前系统的响应时间等数据，确定

最终我们需要达到的响应时间和系统资源使用率等目标；比如：

①登录请求到登录成功的页面响应时间不能超过2秒；

②报表审核提交的页面响应时间不能超过5秒；

③文件的上传、下载页面响应时间不超过8秒；

④服务器的CPU平均使用率小于70%，内存使用率小于75%；

⑤各个业务系统的响应时间和服务器资源使用情况在不同测试环境下，各指标随负载变化的情况等；

**4、制定测试计划的实施时间**

预设本次性能测试各子模块的起止时间，产出，参与人员等等。

**三、测试脚本设计与开发**

性能测试中，测试脚本设计与开发占据了很大的时间比重。

**1、测试环境设计**

本次性能测试的目标是需要验证系统在实际运行环境中的性能外，还需要考虑到不同的硬件配置是否会是制约系统性能的重要因素！因此在测试环境中，需要部署多个不同的测试环境，

在不同的硬件配置上检查应用系统的性能，并对不同配置下系统的测试结果进行分析，得出最优结果（最适合当前系统的配置）。

这里所说的配置大概是如下几类：

①数据库服务器

②应用服务器

③负载模拟器

④软件运行环境，平台

测试环境测试数据，可以根据系统的运行预期来确定，比如需要测试的业务场景，数据多久执行一次备份转移，该业务场景涉及哪些表，每次操作数据怎样写入，写入几条，需要多少的

测试数据来使得测试环境的数据保持一致性等等。

可以在首次测试数据生成时，将其导出到本地保存，在每次测试开始前导入数据，保持一致性。

**2、测试场景设计**

通过和业务部门沟通以及以往用户操作习惯，确定用户操作习惯模式，以及不同的场景用户数量，操作次数，确定测试指标，以及性能监控等。

**3、测试用例设计**

确认测试场景后，在系统已有的操作描述上，进一步完善为可映射为脚本的测试用例描述，用例大概内容如下：

用例编号：查询表单\_xxx\_x1（命名以业务操作场景为主，简洁易懂即可）

用例条件：用户已登录、具有对应权限等。。。

操作步骤：

①进入对应页面。。。。。。

②查询相关数据。。。。。。

③勾选导出数据。。。。。。

④修改上传数据。。。。。。

PS：这里的操作步骤只是个例子，具体以系统业务场景描述；

**4、脚本和辅助工具的开发及使用**

按照用例描述，可利用工具进行录制，然后在录制的脚本中进行修改；比如参数化、关联、检查点等等，最后的结果使得测试脚本可用，能达到测试要求即可；

**PS：**个人而言，建议尽量自己写脚本来实现业务操作场景，这样对个人技能提升较大；一句话：能写就绝不录制！！！

**四、测试执行与管理**

在这个阶段，只需要按照之前已经设计好的业务场景、环境和测试用例脚本，部署环境，执行测试并记录结果即可。

**1、建立测试环境**

按照之前已经设计好的测试环境，部署对应的环境，由运维或开发人员进行部署，检查，并仔细调整，同时保持测试环境的干净和稳定，不受外来因素影响。

**2、执行测试脚本**

这一点比较简单，在已部署好的测试环境中，按照业务场景和编号，按顺序执行我们已经设计好的测试脚本。

**3、测试结果记录**

根据测试采用的工具不同，结果的记录也有不同的形式；现在大多的性能测试工具都提供比较完整的界面图形化的测试结果，当然，对于服务器的资源使用等情况，可以利用一些计数器或

第三方监控工具来对其进行记录，执行完测试后，对结果进行整理分析。

**五、测试分析**

**1、测试环境的系统性能分析**

根据我们之前记录得到的测试结果（图表、曲线等），经过计算，与预定的性能指标进行对比，确定是否达到了我们需要的结果；如未达到，查看具体的瓶颈点，然后根据瓶颈点的具体数据，

进行具体情况具体分析（影响性能的因素很多，这一点，可以根据经验和数据表现来判断分析）。

**2、硬件设备对系统性能表现的影响分析**

由于之前设计了几个不同的测试环境，故可以根据不同测试环境的硬件资源使用状况图进行分析，确定瓶颈是再数据库服务器、应用服务器抑或其他方面，然后针对性的进行优化等操作。

**3、其他影响因素分析**

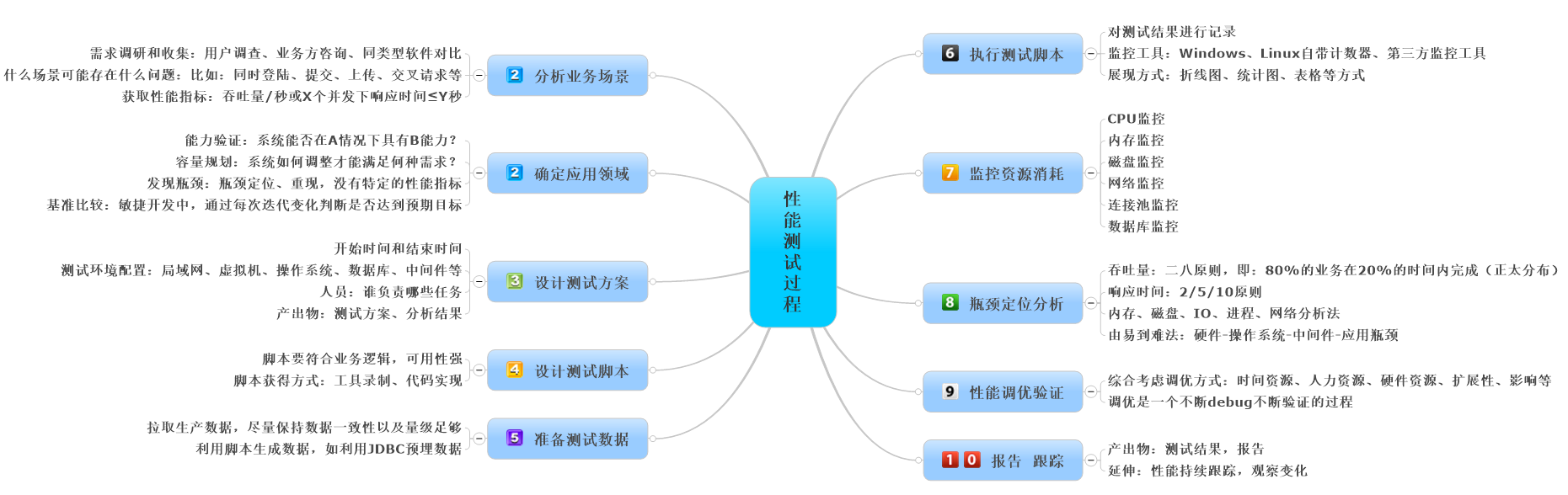
影响系统性能的因素很多，可以从用户能感受到的场景分析，哪里比较慢，哪里速度尚可，这里可以根据2\5\8原则对其进行分析；

至于其他诸如网络带宽、操作动作、存储池、线程实现、服务器处理机制等一系列的影响因素，具体问题具体分析，这里就不一一表述了。

**4、测试中发现的问题**

在性能测试执行过程中，可能会发现某些功能上的不足或存在的缺陷，以及需要优化的地方，这也是执行多次测试的优点。

**六、性能测试思维导图**

****

以上就是一个较简单，完整的性能测试过程，

**七、性能测试的思维导图**

