



## 一、性能需求分析：

性能需求分析是整个性能测试工作开展的基础，如果连性能的需求都没弄清楚，后面的性能测试执行其实是没有任何意义的，而且性能需求分析做的好不好直接影响到性能测试的结果。

一些性能测试人员常犯的错误就是测试一开始就直接用工具对系统进行加压，没有弄清楚性能测试的目的，稀里糊涂做完了以后也不知道结果是否满足性能需求。市面上的书籍也大都是直接讲性能测试工具如LR，jmeter如何使用，导致很多新手一提到性能测试就直接拿工具来进行录制回放，使得很多人认为会使用性能测试工具就等于会性能测试了，殊不知工具其实只是性能测试过程中很小的一部分。

在需求分析阶段，测试人员需要与项目相关的人员进行沟通，收集各种项目资料，对系统进行分析，建立性能测试数据模型，并将其转化为可衡量的具体性能指标，确认测试的目标。所以性能测试需求分析过程是繁杂的，需要测试人员有深厚的性能理论知识，除此之外还需要懂一些数学建模的知识来帮助我们建立性能测试模型。

首先，让我们来看看通过性能需求分析我们需要得出哪些结论或目标：

**明确到底要不要做性能测试？性能测试的目的是什么？**

**明确被测系统是什么？被测试系统的相关技术信息如：架构、平台、协议等**

**明确被测系统的基本业务、关键业务，用户行为**

**明确性能测试点是什么？哪些需要测，为什么？哪些不需要测，又是为什么？**

**明确被测系统未来的业务拓展规划以及性能需求？**

## 明确性能测试策略，即应该怎么测试？

（策略一般如下：

基准测试

压力测试

负载测试

阶梯式测试

稳定性测试）

明确性能测试的指标，知道测试出来的结果怎么算通过？

其次，需求分析阶段我们可以从以下几个方面入手：

### 1、系统信息调研：

指对被测试系统进行分析，需要对其有全面的了解和认识，这是我们做好性能测试的前提，而且在后续进行性能分析和调优时将会大有用处，试想如果连系统的架构、协议都不了解，我们如何进行准确的性能测试？如果进行性能分析与调优？

需要分析的系统信息如下（包括但不限于如下这些）：

### 2、业务信息调研：

指对被测试的业务进行分析，通过对业务的分析和了解，方便我们后续进行性能测试场景的确定以及性能测试指标的确定。

需要分析的业务信息如下（包括但不限于如下这些）：

### 3、性能需求评估：

在实施性能测试之前，我们需要对被测系统做相应的评估，主要目的是明确是否需要做性能测试。如果确定需要做性能测试，需要进一步确立性能测试点和指标，明确该测什么、性能指标是多少，测试通过or不通过的标准？性能指标也会根据情况评估，要求被测系统能满足将来一定时间段的业务压力。

判断是否进行性能测试主要从下面两个方面进行思考：

业务角度：

系统是公司内部 or 对外？系统使用的人数的多少？如果一个系统上线后基本没几个人使用，无论系统多大，设计多么复杂，并发性的性能测试都是没必要的，前期可以否决。当然，除非在功能测试阶段发现非常明显的性能问题，使得用户体验较差的，此时可进行性能测试来排查问题。

系统角度：系统又可以从以下3个方面进行分析

#### a) 系统架构：

如果一个系统采用的框架是老的系统框架（通常大公司都有自己的统一框架），只是在此框架上增加一些应用，其实是没有必要做性能测试，因为老框架的使用肯定是经过了验证的。如果一个系统采用的是一种新的框架，可以考虑做性能测试。

#### b) 数据库要求：

很多情况下，性能测试是大数据量的并发访问、修改数据库，而瓶颈在于连接数据库池的数量，而非数据库本身的负载、吞吐能力。这时，可以结合DBA的建议，来决定是否来做性能测试。

#### c) 系统特殊要求：

从实时性角度来分析，某些系统对响应时间要求比较，比如证券系统，系统的快慢直接影响客户的收益，这种情况就有作并发测试的必要，在大并发量的场景下，查看这个功能的响应时间。

从大数据量上传下载角度分析，某些系统经常需要进行较大数据量的上传和下载操作，虽然此种操作使用的人数不会太多，但是也有必要进行性能测试，确定系统能处理的最大容量，如果超过这个容量时系统需要进行相关控制，避免由于不人工误操作导致系统内存溢出或崩溃。

#### 4、确定性能测试点：

在上面第3点中，我们简单分析了如何确定一个系统是否需要做性能测试。下面简单总结下如果一个系统确定要做性能测试，我们如何确定被测系统的性能测试点？

我们可以从下面几个方面进行分析：

##### 关键业务：

确定被测项目是否属于关键业务，有哪些主要的业务逻辑点，特别是跟交易相关的功能点。例如转账，扣款等接口。如果项目（或功能点）不属于关键业务（或关键业务点），则可转入下面。

##### 日请求量：

确定被测项目各功能点的日请求量（可以统计不同时间粒度下的请求量如：小时，日，周，月）。如果日请求量很高，系统压力很大，而且又是关键业务，该项目需要做性能测试，而且关键业务点，可以被确定为性能点。

##### 逻辑复杂度：

判定被测项目各功能点的逻辑复杂度。如果一个主要业务的日请求量不高，但是逻辑很复杂，则也需要通过性能测试。原因是，在分布式方式的调用中，当某一个环节响应较慢，就会影响到其它环节，造成雪崩效应。

##### 运营推广活动：

根据运营的推广计划来判定待测系统未来的压力。未雨绸缪、防患于未然、降低运营风险是性能测试的主要目标。被测系统的性能不仅能满足当前压力，更需要满足未来一定时间段内的压力。因此，事先了解运营推广计划，对性能点的制定有很大的作用。例如，运营计划做活动，要求系统每天能支撑多少 PV、多少 UV，或者一个季度后，需要能支撑多大的访问量等等数据。当新项目（或功能点）属于运营重点推广计划范畴之内，则该项目（或功能点）也需要做性能测试。

以上 4 点，是相辅相成、环环相扣的。在实际工作中应该具体问题具体分析。例如，当一个功能点不满足以上 4 点，但又属于资源高消耗（内存、CPU），也可列入性能测试点行列。

#### 5、确定性能指标：

性能需求分析一个很重要的目标就是需要确定后期性能分析用的性能指标，性能指标有很多，可以根据具体项目选取和设定，而具体的指标值则需要根据业务特点进行设定，本文不详细进行阐述，后续可考虑就此单独写一篇。

## 二、性能测试准备

### 1、测试环境准备：

a) 系统运行环境：这个通常就是我们的测试环境，有些时候需求比较多，做性能测试担心把环境搞跨了影响其它的功能测试，可能需要重新搭建一套专门用来做性能测试的环境。

b) 执行机环境：这个就是用来生成负载的执行机，通常需要在物理机上运行，而物理机又是稀缺资源，所以我们每次做性能测试都需要提前做好执行机环境。

2、测试场景设计：根据性能需求分析来设计符合用户使用习惯的场景，场景设计的好坏直接影响到性能测试的效果。

3、性能工具准备：

a) 负载工具：根据需求分析和系统特点选择合适的负载工具，比如LR、Jmeter或galting等

b) 监控工具：准备性能测试时的服务器资源、JVM、数据库监控工具，以便进行后续的性能测试分析与调优。

4、测试脚本准备：如果性能测试工具不能满足被测系统的要求或只能满足部分要求时，需要我们自己开发脚本配合工具进行性能测试。

5、测试数据准备：

a) 负载测试数据：并发测试时需要多少数据？比如登录场景？

b) DB数据量大小：为了尽量符合生产场景，需要模拟线上大量数据情况，那么要往数据库里提前插入一定的数据量。这可能需要花费一些时间，特点是关联系统较多，逻辑复杂的业务可能同时涉及多张表。

6、其它：如果需要其它其它关联系统或专业人士如DBA配合的，也需要提前进行沟通。

### 三、性能测试执行

1、人工边执行边分析

通常我们做性能测试都是人工执行并随时观察系统运行的情况、资源的使用率等指标。性能测试的吸引力之一就在于它的不可预知性。当我们在做性能测试的时候遇到跟预期不符的情况很正常，这个时候需要冷静的分析。但这个过程可能会很漫长，需要不断的调整系统配置或程序代码来定位问题，耗时耗人力。特别是在当前敏捷开发模式比较流行的大环境下，版本发布非常频繁且版本周期短（通常1~2周一个版本），没有那么长的时间来做性能测试。

2、无人值守执行性能测试

无人值守是最理想化的目标，目前我们也朝着这个方向努力。无人值守不是说没有人力介入，而是把人为的分析和执行过程分离，执行过程只是机器服从指令的运行而已。通常测试环境在白天比较繁忙，出现性能问题及定位难度较大且会影响功能测试。所以一般性能测试最好在晚上或周末进行，在相对较安静的条件有利于测试结果的稳定性。这种方法也相对比较适合敏捷的模式，不需要人工一直守着。只需要在拿到结果后进行分析就好了。同时，这种方式对测试人员能力的要求比较高，我们需要能进行自动化的收集各种监控数据、生成报表便于后续分析。

### 四、结果分析与调优

关于性能分析与调优这是一个比较大的话题，后续会单独进行总结和分析。

### 五、测试报告与总结

性能测试报告是性能测试的里程碑，通过报告能展示出性能测试的最终成果，展示系统性能是否符合需求，是否有性能隐患。性能测试报告中需要阐明性能测试目标、性能测试环境、性能测试数据构造规则、性能测试策略、性能测试结果、性能测试调优说明、性能测试过程中遇到的问题和解决办法等。

性能测试工程师完成该次性能测试后，需要将测试结果进行备案，并做为下次性能测试的基线标准，具体包括性能测试结果数据、性能测试瓶颈和调优方案等。同时需要将测试过程中遇到的问题，包括代码瓶颈、配置项问题、数据问题和沟通问题，以及解决办法或解决方案，进行知识沉淀。