# 性能测试方案设计思路总结

By：授客 QQ：1033553122

博客：http://blog.sina.com.cn/ishouke

欢迎加入软件性能测试交流 QQ 群：7156436



## 一、 需求分析

### 1． 测试目的

为什么测？目的在于测试系统相关性能能否满足业务需求。通常分以下两种情况：

1）新项目上线

2）老项目优化

如果是老项目优化，可考虑是否存有历史测试方案，如果有可以参考，或许可以省事很多。

### 2． 测试对象

**要测啥？**

测试对象可以归结为“业务功能”。测试前，需要了解我们需要测试的业务功能（不深入细节）有哪些，比如“购买商品”、“寄送快递”。

**有没有必要测？**

需求来源哪里？，有没有数据支撑测试这个需求的必要性？

通常，可以从以下几个方面考虑：

1）是否核心功能，是否要求严格的质量

2）是否常用、高频使用的功能

3）可能占用系统较多资源的功能

4）使用人数多还是少

5）在线人数多还是少

### 3． 拆分对象

先从业务上来分，实现这个完整的功能包含哪些流程、环节

举例：购买商品

登录->搜索商品->提交订单->支付订单->退出

然后从功能实现上来看，怎么实现这个完整功能的。通常这些业务功能操作都对应着一个或多个请求(可能能是不同类型的请求，比如 http, mysql 等)，我们要做的是找出这些操作对应的请求，请求之间的顺序是怎么样的。

### 4． 指标分析

分析性能需求指标（如“支持 300 人并发登录”）是否合理

有必要测试这个需求，考虑需求指标是否合理？有没有数据支撑？

通常，支撑数据可以从以下方面考虑：

1）采样时间段内系统使用人数

2）采样时间段内系统在线人数

3）采样时间段内系统(页面)访问量

4）采样时间段内请求数

....

常用分析思路：

**1）2/8 法则**

2/8 法则：80%的业务量在 20%的时间里完成。这里，业务量泛指访问量，请求数，数据量等

**2）正态分布**

**3）按比例倍增**

**4）响应时间 2-5-8 原则**

就是说，一般情况下，当用户能够在 2 秒以内得到响应时，会感觉系统的响应很快；当用户在 2-5 秒之间得到响应时，会赶紧系统的响应速度还可以；当用户在 5-8 秒以内得到响应时，会赶紧系统的速度很慢，但是还可以接受；而当用户在超过 8 秒后仍然无法得到响应时，会感觉系统糟糕透了，或者认为系统已经失去响应。

注意：这个要根据实际情况，有些情况下时间长点也是可以接受的，好比 12306

举例：

某公司后台监控，根据一段时间的采样数据，分析得出日高峰时段(11:00-14:00)用户下单请求数平均为1000，峰值为1500，根据这个计算并发请求数

时段：3 个小时 -> 3 x 60 x 60 = 1080s

业务量：1500

吞吐量：1500 * 80% / (1080 * 20%) = 5.56 请求数/s

假设用户下单遵循正态分布，那么并发请求数峰值会肯定大于上述估算的吞吐量

注意：

1、2/8 原则计算的结果并非在线并发用户数，是系统要达到的处理能力（吞吐量）

2、如果要求更高系统性能，根据实际情况，也可以考虑 1/9 原则或其它更严格的算法

3、以上估值只是大致的估算，不是精确值

举例：

想了下，暂时没想到啥好的例子，大致就说一些涉及到数据量的性能测试，比如报表统计，或者是大数据挖掘，查询等，怎么去估算数据量？

数据生命周期：

一般来说，数据都是有一定的生命周期的，时间的选取需要结合数据周期考虑。这里假设 3 年后系统性能仍然需要满足业务需求。

数据增长率：

如果是老项目，可以考虑对应功能主表历史数据存放情况

这里假设按年统计，比如第一年 10000，第二年 15000，第三年 20000，第四年 25000，那么我们得出，以第一年为基准，数据增长率分别为 0.5，1，1.5，每年在上一年的基础上，以 5000 的速度增长

预估 3 年后，数据增长率为 3，需要测试数据量为 （1+3）x 10000 = 40000

注意：

1、实际数据一般是没上面举例那么规律的，只能大致估算数据增长率。

2、一些大数据量的性能测试除了和数据量相关，还涉及到数据分布等，比如查询，构造数据时需要结合实际，尽量贴近实际。

3、不同业务模块，涉及表不一样，数据量要求也是不一样的，需要有区别的对待。

如果是新项目，那就比较不确定了，除非能收集相关数据。

## 二、 系统分析

结合需求分析中第 3 点，分析系统架构。

1）请求顺序、请求之间相互调用关系

2）数据流向，数据是怎么走的，经过哪些组件、服务器等

3）预测可能存在性能瓶颈的环节（组件、服务器等）

4）明确应用类型 IO 型，还是 CPU 消耗性、内存消耗型 -> 弄清楚重点监控对象

5）关注应用是否采用多进程、多线程架构 -> 多线程容易造成线程死锁、数据库死锁，数据不一致等

6）是否使用集群/是否使用负载均衡

了解测试环境部署和生产环境部署差异，是否按 1:1 的比例部署

通常建议测试时先不考虑集群，采用单机测试，测试通过后再考虑使用集群，这样有个比较，比较能说明问题

参考阅读“浅谈 web 网站架构演变过程 ”：
http://blog.csdn.net/qiaqia609/article/details/50809383

## 三、 业务分析

1）明确要测试的功能业务中，功能业务占比，重要程度。

目的在于

<1>明确重点测试对象，安排测试优先级

<2>建模，混合场景中，虚拟用户资源分配，针对不同业务功能施加不同的负载。

2）明确下“需求分析-指标分析”中相关业务功能所需基础数据及数据量问题，因为那块需求分析时可能只是大致估算下，评估指标是否合理，需要认真再分析下

## 四、 用例设计

### 1）用例设计

通常是基于场景的测试用例设计

<1> 单业务功能场景

运行测试期间，所有虚拟用户只执行同一种业务功能某个环节、操作

<2> 混合业务功能场景

运行测试期间，部分虚拟用户执行某种业务的某个环节操作，部分虚拟用户执行该业务功能的其它环节

或者

运行测试期间，部分虚拟用户执行某种业务功能，部分虚拟用户执行其它业务功能

注：这里用例没说到多少用户去跑，跑多久等，这里只是把他当作相同场景用例下的的一组组测试数据了。

**2）事务定义**

根据用例合理的定义事务，方便分析耗时（特别是混合业务功能场景测试），进而方便分析瓶颈。

比如，购买商品，我们可以把下订单定义为一个事务，把支付也定义为一个事务。

**3）场景监控对象**

针对每条用例，结合“系统分析”第 4）点，明确可能的压力点（比如数据库、WEB 服务器），需要监控的对象，比如 tps，耗时，CPU，内存，I/O 等

**五、 测试策略**

1）先进行混合业务功能场景的测试，在考虑进行测试单业务功能场景的测试

2）负载测试 -> 压力测试-> 稳定性测试-> 强度测试

注：如果测试稳定性，时间建议至少 8 小时；

3）逐步加压

比如开始前 5 分钟，20 个用户，然后每隔 5 分钟，增加 20 个用户。

好处：不仅比较真实的模拟现实环境，而且在性能指标比较模糊，且不知道服务器处理能力的情况下，可以帮我们确定一个大致基准，因为通常情况下，随着用户数的不断增加，服务器压力也会随着增加，如果服务器不够强大，那么就会出现不能及时处理请求、处理请求失败的情况下，对应的运行结果图形中，运行曲线也会出现对应的形态，比如从原本程一条稳定直线的情况，到突然极限下降、开始上下波动等，通过分析我们就能得出服务器大致处理能力，供后续测试参考。

4）单点并发

比如使用集合点，单独针对某个环节的并发测试，通常是针对某个环节的性能调优时使用。

常识：

a) 负载测试

保证系统能正常运行(通常是满足某些系统性能指标)的前提下，让被测对象承担不同的工作量，以评估被测对象的最大处理能力及存在缺陷而进行的测试

b) 压力测试

不保证系统能否正常运行的前提下，让被测对象承担不同工作量，以评估被测对象能提供的最大处理能力及存在缺陷而进行的测试

c）稳定性测试

测试系统的长期稳定运行的能力。同疲劳强度测试的区别是，稳定性测试的压力强度较小，一般趋向于客户现场日常状态下的压力强度，当然在通过时间不能保证稳定性的状态下，需要加大压力强度来测试，此时的压力强度则会高于正常值。

d）强度测试

通常模拟系统在较差、异常资源配置下运行，如人为降低系统工作环境所需要的资源，如网络带宽，系统内存，数据锁等等，以评估被测对象在资源不足的情况下的工作状态

注：疲劳强度测试是一类特殊的强度测试，主要测试系统长时间运行后的性能表现，例如 7x24 小时的压力测试。

## 六、 工具选取

### 1）协议分析

一般性能测试工具都是基于协议开发的，所以先要明确应用使用的协议

### 2）工具选取

1）类型

开源工具、收费工具、自研工具

2）分析工具

<1> 理解工具实现原理

<2> 采用用异步还是同步

常识：

1. 同步请求：发出一个调用请求，在没有得到结果之前，该调用就不返回。

2.异步请求：发出一个调用请求，在没有得到请求结果之前，该调用可立即返回。该调用请求的处理者在处理完成后通过状态、通知和回调等来通知调用者。

<3> 使用长连接还是短连接

## 七、 软件配置

1）操作系统

内核版本、32 or 64 位?

2）应用版本

应用版本要和线上保持一致，特别是中间件、组件等的版本，因为不同版本，其性能可能不一样

3）参数配置

<1> 负载均衡、反向代理参数配置

<2> Web 服务器参数配置

<3> 数据库服务器参数配置

## 八、 网络分析

1）网络路由

通常为了排除网络型瓶颈，通常建议在局域网下进行测试。

通常，这里我的分析思路是这样的：

<1> 检查 hosts 文件的配置

从终端压测机(负载生成机)开始，到请求目的服务器器，机器的 hosts 文件配置

通常，hosts 文件位于如下：

Windows：C:\Windows\System32\drivers\etc\hosts

Unix/Linux：/etc/hosts

小常识：

1、通常域名访问站点，首先要通过 DNS 域名服务器把网络域名（形如 www.xxx.com）解析成 XXX.XXX.XXX.XXX 的 IP 地址，然后继续后续访问。

2、hosts 存放了域名和 ip 地址的映射关系，如下

```
# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost

192.168.  .1   s.test.    cn
192.168.  .1   hop.      fu.cn
192.168.  .1   waimai.      u.cn
192.168.  .    mkt.1      .cn
192.168.  .    s.         .cn
192.168.  .    hop.       .cn
```

使用 hosts 可以加快域名解析，在进行 DNS 请求以前，系统会先检查自己的 hosts 文件中是否有这个地址映射关系，如果有则把域名解析为映射的 IP 地址，不请求网络上的 DNS 服务器，如果没有再向已知的 DNS 服务器提出域名解析。也就是说 hosts 的请求级别比 DNS 高，可加快域名解析。

<2> 检查 DNS 配置

不同 DNS，其速度和准确率是不一样的，比如 114.114.114.114 速度远比 8.8.8.8 快，如果有用到 DNS（特别是压测机），需要考虑下是否适当

<3> 确保路由正确设置

2）网络带宽

如果没条件在局域网下测试，可能需要估算所需大致带宽。

如果测试时是基于 UI 层操作的操作，那么得估算页面平均大小，这个可以通过浏览器自带工具查看打开单个页面服务器返回的请求数据大小。如果是测试时是基于接口层的请求测试，可以通过工具查看服务器响应数据大小。

然后根据采集的页面 PV 峰值、请求数峰值进行计算。

假设在 PV 峰值、请求数峰值 = 1000，峰值时段：8:00 - 12:00，平均页面、请求大小 200k

带宽 = 1000 x 80% / (20% x 4 x 3600s) x 200KB x /1024 x 8bit ,单位 MBps

注意： 这里涉及到浏览器缓存等因素，估值可能不准，大致估算。

**九、 硬件配置**

1) CPU

型号，频率，核数

2) 内存

3) 磁盘

不同磁盘类型，读写速率不一样

4) 网卡

不同网卡，其传输速率也不一样

注意：硬件配置最好和生产环境的配置保持一致

**十、 性能监控**

略

注意：

1） 这里监控不仅仅是服务器自身性能指标监控，如 cpu，还包括事务耗时监控等

2） 需要记录测试前各个性能指标数据，方便后续测试对比

**十一、 实施测试**

略

## 十二、 结果分析

如果是性能调优，还需同上一个版本的性能测试结果对比

略