第九章节 性能测试概念篇

目标：能够对个人编写的项目进行接口的性能测试。

1.什么是性能测试?

> 性能测试和功能测试都是在系统测试阶段进行，那么这两者有什么区别呢?

案例1：五菱和法拉利都是汽车厂商生产的汽车，从功能上说，它们都有四个轮子一个方向盘，能够坐在里面往前开，有挡风玻璃能够遮风挡雨。

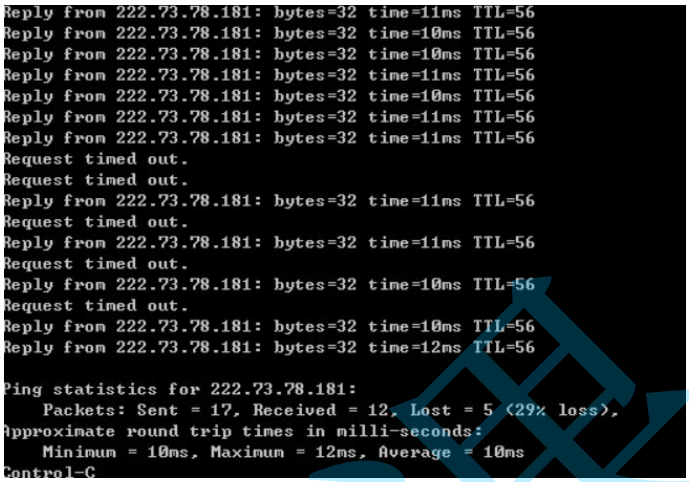
从性能来说，五菱的座椅材质可能是人造皮，法拉利是真皮；速度从0-100km/h，五菱可能需要十几秒，而法拉利只需要几秒，这就是性能的区别。

对于一个事情来说，能做是一回事，做的好不好又是另一回事情。就跟做饭一样，能不能吃是一回事，好不好吃又是另一回事。

概念：为了发现系统性能问题或获取系统性能相关指标而进行的测试。

一般在真实环境、特定负载条件下，通过工具模拟实际软件系统的运行及其操作，同时监控性能各项指标，最后对测试结果进行分析来确定系统的性能情况。

对于软件来说什么是性能问题：



> 以购物软件为例：
>
> 1）购物过程中页面突然无法打开，刷新后可以重新打开
>
> 2）双十一期间无法进入商品页面
>
> 3）页面加载时间过长，需要消耗用户大量的等待时间
>
> ……

常见的性能问题：

查询数据时间过长，网速很慢，服务器无响应，查询数据很长时间才显示列表。

2.常见性能测试指标

如何来评估系统性能的好坏？需要借助性能指标来统计和分析。

并发数

并发用户数。

从业务层面看，并发用户数指的是实际使用系统的用户总数。

从后端服务器层面看，指的是web服务器在一段时间内处理浏览器请求而建立的http连接数或生成的处理线程数。

> 案例：一个已经投入运行的web系统，有5000名员工使用，最多同时有2500人使用，用户分别进行浏览页面、填写订单、提交订单、查询订单的操作。那么这个系统的业务并发用户数是2500，实际并发用户数是提交订单和查询订单操作的用户。

吞吐量

单位时间内处理的并发数，直接体现软件系统负载承受能力。吞吐量越高，系统承受的并发越多，性能越好。

> 案例：

> 有AB两种场景。
>
> A场景，有100个并发用户，每个用户每隔1秒发出一个请求。
>
> B场景，有1000个并发用户，每个用户每隔10秒发送一个请求。
>
> A和B场景的吞吐量相同都是每秒100个请求。但是A场景思考时间短，所以A场景占用的系统资源更多。

吞吐量分类：

1）按照请求数量：TPS和QPS

`TPS`：每秒处理事务数，用于衡量系统在一定时间内能够处理的事务数

计算公式：总的请求成功的事务数 / 总的运行时间

举例1：某一系统1分钟处理1000个业务，那么TPS = 1000 / 60 = 16.7

举例2：2022年最高的一天又10万笔交易，预测2023年TPS需要多少合格？认为每笔交易就是一个事务，理论TPS = 100000 / 24*60*60 = 1.2（理想状态），然而实际上订单量会在某段时间内突然增加，并不是平均到每个时间段内，因此

1）没有更详细的数据：根据二八定律（80%的事务在20%的时间内完成）

TPS = 100000 * 0.8 / 24*60*60*0.2 = 4.6

2）如果有详细的数据：5万笔交易在晚上的8~9点完成的

TPS = 50000 / 60*60 = 13.9（实际还要参考往年业务的增长，假设每年业务增长30%，则TPS = 50000 + 50000*0.3 / 60*60 = 18）

> 一般来说，吞吐量越大，性能越好

`QPS`：每秒查询率

若一个事务中只有一个接口且是查询接口，则QPS = TPS；

2）按照网络数据包划分：KB

响应时间

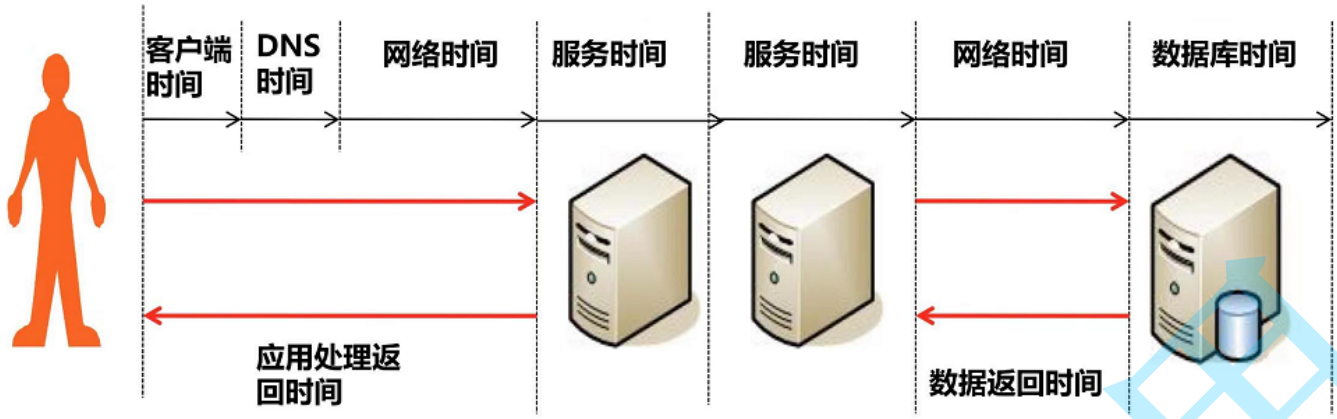
验证系统处理速度快不快。

应用系统从请求发出开始，到客户端接收到最后一个字节数据所消耗的时间。

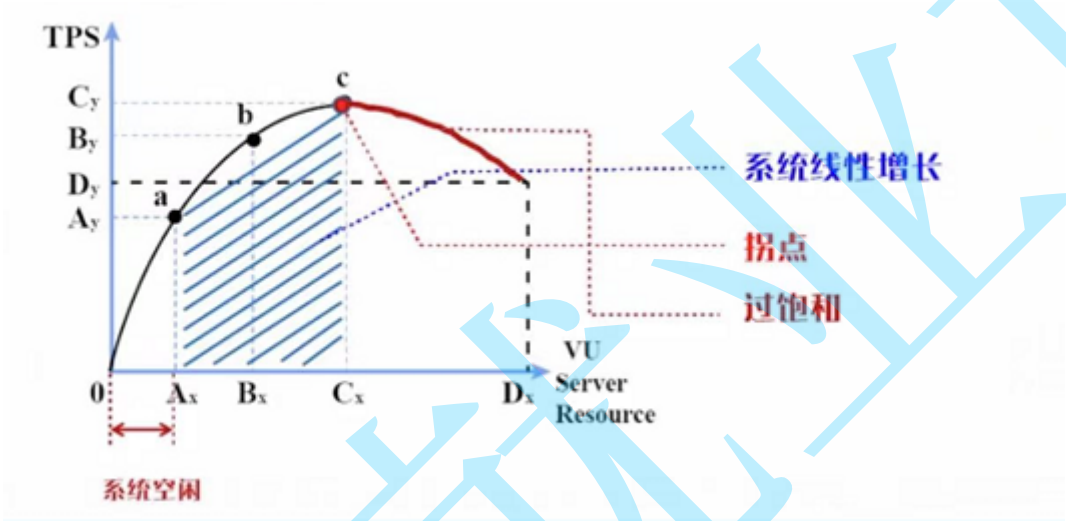
对于web系统而言，系统响应时间包含前端展现时间和系统响应时间。

前端展现时间：页面渲染时间。

系统响应时间：包含服务器、数据库、通讯网络等响应时间。



> 并发用户、系统吞吐量、系统响应时间之间的关系



当并发用户较少，系统吞吐量低，系统响应时间较短，我们认为系统处于空闲区间。随着系统并发用户增加，系统吞吐量开始呈线性增长，系统性能进入了线性增长区间。

吞吐量在某个点上达到了饱和点，也称之为拐点。在这之后用户请求不再被立即处理，响应时间随之变长，吞吐量也逐渐降低，系统性能进入了过饱和区间。

系统性能的拐点通常是性能测试的主要目的。

资源利用率

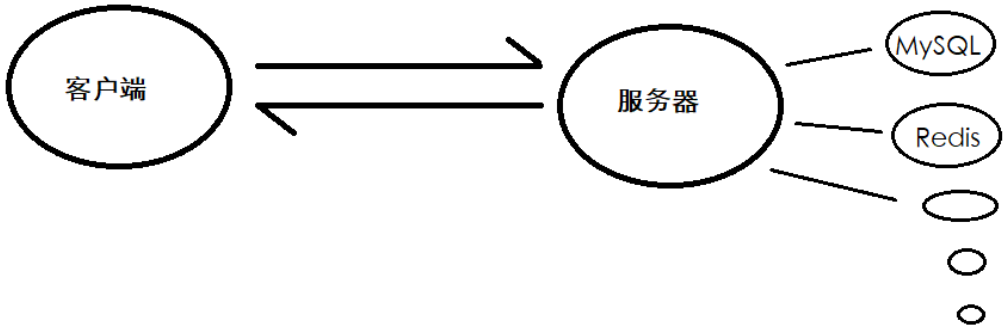
通过查看系统占用的情况分析资源瓶颈。

服务器：CPU、内存、磁盘、网络等

3.性能测试关注点

不同的角色看待性能测试的侧重点不同。

从从客户端发起一个请求到客户端收到请求的整个过程中，各阶段都可能存在性能问题。



> *后端处理请求的性能问题*
>
> *服务器硬件资源（CPU、内存、磁盘）*
>
> *中间件、网络、数据库、架构设计等是否存在瓶颈。*

开发人员和测试人员并不需要关注全流程的性能问题。通过分析不同角色的侧重点，从而促进性能测试更好的开展。和软件系统性能相关的角色主要有四种。

- 终端用户

  对于终端用户来说，表现为用户进行业务操作时的主观响应时间。用户重点关注从你提交请求到收到响应的时间，包括系统响应时间和前端展现时间。

- 系统运维人员

  系统运维人员除了关注单个请求的响应时间，更关注大量用户并发访问时对系统的影响，以及更大负载情况下的系统健康状态。从而执行系统的整体的策略。

  > 1.关注全局利益最大化，对最大并发用户数和系统响应时间进行权衡取舍
  >
  > 2.系统并发处理时间、系统容量、数据库调优，以及长时间运行稳定性和可扩展性。
  >
  > 例如：当有以下两种方案时
  >
  > A方案：100万并发访问用户，登陆响应时间3秒
  >
  > B方案：500万并发访问用户，登陆响应时间8秒
  >
  > 这种情况下运维人员往往更更倾向于第二种。

- 软件设计开发人员

  关注算法设计、架构设计、性能最佳实践、数据库相关、软件性能的可测试性等方面。

  > 例如：对于算法，要保证高效，无内存泄漏；对于架构，要保证系统容量和性能可扩展。

- 性能测试人员

  工作重点在于性能测试场景的设计、脚本的开发和执行，以及性能缺陷的排查和定位。

  测试人员除了具有及其宽广的知识面，如系统架构，存储架构，网络架构等全局的知识，还要有大量知识积累，比如数据库SQL语句的执行计划调优、JVM垃圾回收、多线程常用问题等。

  > 可见性能测试的范围太大了，不同的角色对于性能有不同的处理方式。

4.性能测试分类

基准测试

> 基准测试(Benchmark Testing)又称单用户测试，主要用于监测被测系统在较低压力下的运行状况并记录相关数据。当性能测试环境确定以后，通常选取业务模型中的重要业务做基准测试，对被测系统施加一定压力，从而获取被测系统在单用户运行情况下的各项性能指标，为多用户并发测试和混合场景测试等提供参考依据

并发测试

> 并发测试(Concurrency Testing)用于评估被测系统的某些特定操作同时发生时的性能表现，例如，被测系统被多个用户同时登录时的响应能力，或系统的某一功能被多个用户同时操作时的性能表现。通过并发测试，不仅可以获得被测系统在多用户并发操作时的性能指标，还可以发现被测系统在并发条件下可能发生的问题，如内存泄漏、线程锁、资源争用问题。例如，通过模拟多个用户同时访问某一条件数据，或模拟多个用户同时更新数据，可能会发现被测系统的数据库访问错误、写入错误等。几乎所有的性能测试都会涉及一些并发测试。但并发测试对并发时间要求比较苛刻，通常需借助专门的性能测试工具，采用多线程或多进程的方式来模拟多个虚拟用户的并发性操作。

负载测试

> 负载测试(Load Testing)是性能测试的一种测试类型，用于评估被测系统在预期的不同负载下的行为。负载测试关注系统处理不同负载的能力，这些负载可通过控制并发用户或者进程的数量来实现。进行负载测试时，通过对系统不断增加并发访问负载，监测系统性能的变化，直到系统的某项或多项性能指标达到安全临界值，最终确定在满足该安全临界值的性能指标下，系统所能承受的最大负载量。简而言之，负载测试是通过逐步加载的方式来确定系统的处理能力。负载测试类似于举重运动，通过不断给运动员增加重量，确定运动员在其身体状况保持正常的情况下所能举起的最大重量。通过负载测试可以获取系统能够达到的峰值指标。

例如，一个软件系统的响应时间要求不超过2秒，如果在这个前提下不断增加用户访问量，系统的响应时间就会变长。假设当访问量超过1万人时系统的响应时间超过2秒，那么就可以确定在系统响应时间不超过2秒的前提下，系统的最大负载量是1万人。负载测试可用于系统的性能验证、性能诊断和性能调优等场景

压力测试

压力测试(Stress Testing)用于评估被测系统在高于预期、高于指定容量负载需求或低于最少需求资源的条件下的行为。压力测试关注被测系统处理超出预期或特定峰值负载的能力，也可以用于评估系统在资源匮乏时的处理能力，比如在可用的计算能力、带宽和内存资源不足的条件下系统的表现。进行压力测试时通常采用逐步增加系统负载的方式，使系统某些资源达到饱和甚至失效，从而发现那些只有在高负载条件下才会出现的缺陷，如同步问题、内存泄漏等。通过对被测系统进行压力测试，也能找出被测系统的性能拐点，获得系统所能提供的最大服务级别（系统所能承受的最大压力），评估系统在峰值负载或超出最大负载情况下的处理能力。压力测试主要用于性能诊断、性能调优和容量规划等场景。

压力测试和负载测试的区别?

压力测试与负载测试不同。负载测试是在保持性能指标要求的前提下测试系统能够承受的最大负载，而压力测试则是测试系统性能达到极限的状态。例如，软件系统要求的响应时间为2秒。进行负载测试时发现，当访问量达到1万时，系统响应时间不超过2秒，而当访问量超过1万时，系统响应时间则会超过2秒，那么，在满足系统响应时间指标的前提下，该系统能够承受的最大访问量是1万。进行压力测试时，则可继续增加系统的访问量，并观察系统的性能变化。例如，当系统访问量增加到2万时，发现系统响应时间延迟到5秒，而当访问量增加到3万时，系统则崩溃，无法做出响应。由此可以确定系统能达到的极限访问量是3万

稳定性测试

> 在负载测试的基础上，执行较长时间的测试以检查系统的稳定性。通常较长时间指3*24小时以上。