常见的[**性能测试**](javascript:;)方法有以下几种：  
1．负载测试  
        在这里，负载测试指的是最常见的验证一般性能需求而进行的性能测试，在上面我们提到了用户最常见的性能需求就是“既要马儿跑，又要马儿少吃草”。因此负载测试主要是考察软件系统在既定负载下的性能表现。我们对负载测试可以有如下理解：  
（1）负载测试是站在用户的角度去观察在一定条件下软件系统的性能表现。  
（2）负载测试的预期结果是用户的性能需求得到满足。此指标一般体现为响应时间、交易容量、并发容量、资源使用率等。  
2．压力测试  
        压力测试是为了考察系统在极端条件下的表现，极端条件可以是超负荷的交易量和并发用户数。注意，这个极端条件并不一定是用户的性能需求，可能要远远高于用户的性能需求。可以这样理解，压力测试和负载测试不同的是，压力测试的预期结果就是系统出现问题，而我们要考察的是系统处理问题的方式。比如说，我们期待一个系统在面临压力的情况下能够保持稳定，处理速度可以变慢，但不能系统崩溃。因此，压力测试是能让我们识别系统的弱点和在极限负载下程序将如何运行。   
        例子：负载测试关心的是用户规则和需求，压力测试关心的是软件系统本身。对于它们的区别，我们可以用华山论剑的例子来更加形象地描述一下。如果把郭靖看做被测试对象，那么压力测试就像是郭靖和已经走火入魔的欧阳峰过招，欧阳锋蛮打乱来，毫无套路，尽可能地去打倒对方。郭靖要能应对住，并且不能丢进小命。而常规性能测试就好比郭靖和黄药师、洪七公三人约定，只要郭靖能分别接两位高手一百招，郭靖就算胜。至于三百招后哪怕郭靖会输掉那也不用管了。他只要能做到接下一百招，就算通过。  
  思考  
        我们在做软件压力测试时，往往要增加比负载测试更多的并发用户和交易，这是为什么？  
3．并发测试  
        验证系统的并发处理能力。一般是和服务器端建立大量的并发连接，通过客户端的响应时间和服务器端的性能监测情况来判断系统是否达到了既定的并发能力指标。负载测试往往就会使用并发来创造负载，之所以把并发测试单独提出来，是因为并发测试往往涉及服务器的并发容量，以及多进程/多线程协调同步可能带来的问题。这是要特别注意，必须测试的。  
4．基准测试  
        当软件系统中增加一个新的模块的时候，需要做基准测试，以判断新模块对整个软件系统的性能影响。按照基准测试的方法，需要打开/关闭新模块至少各做一次测试。关闭模块之前的系统各个性能指标记下来作为基准（Benchmark），然后与打开模块状态下的系统性能指标作比较，以判断模块对系统性能的影响。  
5．稳定性测试  
        “路遥知马力”，在这里我们要说的是和性能测试有关的稳定性测试，即测试系统在一定负载下运行长时间后是否会发生问题。软件系统的有些问题是不能一下子就暴露出来的，或者说是需要时间积累才能达到能够度量的程度。为什么会需要这样的测试呢？因为有些软件的问题只有在运行一天或一个星期甚至更长的时间才会暴露。这种问题一般是程序占用资源却不能及时释放而引起的。比如，内存泄漏问题就是经过一段时间积累才会慢慢变得显著，在运行初期却很难检测出来；还有客户端和服务器在负载运行一段时间后，建立了大量的连接通路，却不能有效地复用或及时释放。  
6．可恢复测试  
        测试系统能否快速地从错误状态中恢复到正常状态。比如，在一个配有负载均衡的系统中，主机承受了压力无法正常[**工作**](javascript:;)后，备份机是否能够快速地接管负载。可恢复测试通常结合压力测试一起来做。  
        提示：每种测试有其存在的空间和目的。当我们接手一个软件项目后，在有限的资源条件下，选择去做哪一种测试，这应该根据当前软件过程阶段和项目的本身特点来做选择。比如，在集成测试的时候要做基准测试，在软件产品每个发布点要做性能测试。