**WEB性能测试方法**

性能测试用例主要分为预期目标用户测试，用户并发测试，疲劳强度与大数据量测试，网络性能测试，服务器性能测试五大部分，具体编写测试用例时要根据实际情况进行裁减，在项目应用中遵守低成本，策略为中心，裁减，完善模型，具体化等原则；  
一、WEB 全面性能测试模型  
Web 性能测试模型提出的主要依据是：一种类型的性能测试可以在某些条件下转化成为另外一种类型的性能测试，这些类型的性能测试的实施是有着相似之处的；  
1. 预期指标的性能测试  
系统在需求分析和设计阶段都会提出一些性能指标，完成这些指标的相关的测试是性能测试的首要工作之一，这些指标主要诸于“系统可以支持并发用户200个；”系统响应时间不得超过20秒等，对这种预先承诺的性能要求，需要首先进行测试验证；  
2. 独立业务性能测试  
独立业务实际是指一些核心业务模块对应的业务，这些模块通常具有功能比较复杂，使用比较频繁，属于核心业务等特点。  
用户并发测试是核心业务模块的重点测试内容，并发的主要内容是指模拟一定数量的用户同时使用某一核心的相同或者不同的功能，并且持续一段时间。对相同的功能进行并发测试分为两种类型，一类是在同一时刻进行完全一样的操作。另外一类是在同一时刻使用完全一样的功能。  
3. 组合业务性能测试  
通常不会所有的用户只使用一个或者几个核心业务模块，一个应用系统的每个功能模块都可能被使用到；所以WEB性能测试既要模拟多用户的相同操作，又要模拟多用户的不同操作；组合业务性能测试是最接近用户实际使用情况的测试，也是性能测试的核心内容。通常按照用户的实际使用人数比例来模拟各个模版的组合并发情况；组合性能测试是最能反映用户使用情况的测试往往和服务器性能测试结合起来，在通过工具模拟用户操作的同时，还通过测试工具的监控功能采集服务器的计数器信息进而全面分析系统瓶颈。  
用户并发测试是组合业务性能测试的核心内容。组合并发的突出特点是根据用户使用系统的情况分成不同的用户组进行并发，每组的用户比例要根据实际情况来匹配；  
4. 疲劳强度性能测试  
疲劳强度测试是指在系统稳定运行的情况下，以一定的负载压力来长时间运行系统的测试，其主要目的是确定系统长时间处理较大业务量时的性能，通过疲劳强度测试基本可以判定系统运行一段时间后是否稳定；  
5. 大数据量性能测试  
一种是针对某些系统存储，传输，统计查询等业务进行大数据量时的性能测试，主要针对某些特殊的核心业务或者日常比较常用的组合业务的测试；  
第二种是极限状态下的数据测试，主要是指系统数据量达到一定程度时，通过性能测试来评估系统的响应情况，测试的对象也是某些核心业务或者常用的组合业务。  
第三种大数据量测试结合了前面两种的测试，两种测试同时运行产生较大数据量的系统性能测试；  
大数据量测试通常在投产环境下进行，并独立出来和疲劳强度测试放在一起，在整个性能测试的后期进行；大数据量的测试可以理解为特定条件下的核心业务或者组合业务测试；  
6. 网络性能测试  
主要是为了准确展示带宽，延迟，负载和端口的变化是如何影响用户的响应时间的，在实际的软件项目中  
主要是测试应用系统的用户数目与网络带宽的关系。网络测试的任务通常由系统集成人员完成；  
7. 服务器（操作系统，WEB服务器，数据库服务器）性能测试  
初级服务器性能测试主要是指在业务系统工作或者进行前面其他种类性能测试的时候，监控服务器的一些计数器信息，通过这些计数器对服务器进行综合性能分析，为调优或提高系统性能提供依据；  
高级服务器性能测试一般由专门的系统管理员来进行如数据库服务器由专门的DBA来进行测试和调优；  
8. 一些特殊的测试  
主要是指配置测试，内存泄露测试的一些特殊的WEB性能测试；  
二、WEB 性能测试策略  
性能测试策略一般从需求设计阶段开始讨论如何定制，它决定着性能测试工作要投入多少资源，什么时间开始实施等后续工作的安排；其制定的主要依据是软件自身的特点和用户对性能的关注程度，其中软件自身的特点起决定性的作用；  
软件按照用途的不同可以分为两大类，系统类软件和应用类软件。系统类软件通常对性能要求较高，因此性能测试应该尽早介入；应用类软件分为特殊类应用和一般类应用，特殊类应用主要有银行，电信，电力，保险，医疗，安全等领域软件，这类软件使用频繁，用户较多，也需要较早进行性能测试；一般类主要是指一些普通类应用如OA，MIS 等一般类软件根据实际情况制定性能测试策略，受用户因素影响较大；  
1. 系统类软件  
从设计阶段就开始针对系统架构，数据库设计等方面进行讨论，从根源来提高性能，系统类软件一般从单元测试阶段开始性能测试实施工作，主要是测试一些和性能相关的算法和模块；  
2. 应用类软件  
特殊应用：从设计阶段就开始针对系统架构，数据库设计等方面进行讨论，从根源来提高性能，系统类软件一般从单元测试阶段开始性能测试实施工作，主要是测试一些和性能相关的算法和模块；  
一般应用：与使用用户的重视程度有关，用户高度重视时，设计阶段开始进行一些讨论工作，主要在系统测试阶段开始进行性能测试实施；用户一般重视时，可以在系统测试阶段的功能测试结束后进行性能测试；用户不怎么重视时，可以在软件发布前进行性能测试，提交测试报告即可；  
三、WEB性能测试用例设计模型  
性能测试用例设计通常不会一次设计到位，是一个不断迭代完善的过程，即使在使用过程中，也不是完全按照设计好的测试用例来执行，需要根据需求的变化进行调整和修改； WEB性能测试用例设计模型是一个内容全面比较容易组织和调整的模型架构。  
1. 预期性能指标测试用例  
指一些十分明确的，在系统需求设计阶段预先提出的，期望系统达到的，或者向用户保证的性能指标，针对每个指标都要编写一个或者多个测试用例来验证系统是否达到要求，预期性能指标测试用例主要参考需求和设计文档，把里面十分明确的性能要求提取出来，指标中通常以单用户为主；  
如：对于普通的客户端，系统上传5MB以内的文件，速度不低于2MB/S；  
输入动作：选择1-5 MB的文件并上传，用秒表计时；  
期望的性能：上传的时间小于等于2.5S  
实际性能：上传的时间2.29秒；  
这类用例通常以手工的方式执行；  
2. 用户并发性能测试用例  
用户并发测试主要通过逐渐增加用户数量来加重系统负担，并通过测试工具对应用系统，各种服务器资源进  
监控，用户并发测试可以是正常数量用户和特殊数量用户进行并发， 用户并发测试是系统性能测试的核心部分，涉及压力测试，负载测试，强度测试等多方面的内容.独立业务性能测试实际就是核心业务模块的某一业务的并发性能测试，可以理解为单元性能测试；组合业务的性能测试是一个或者多个模块的多个业务同时进行并发性能测试，可以理解为集成性能测试，单元性能测试和集成性能测试两者紧密相连合并称为用户并发性能测试；用户并发测试要求选择有代表性的关键的业务来设计测试用例，以便更有效的评测系统性能；其测试用例设计文档的基本的编写思想是按照系统的体系结构进行编写.  
3. 独立核心模块用户并发性能的测试用例设计  
完全一样功能的并发测试：主要检查系统的健壮性，从技术角度讲就是检查程序对同一时刻并发操作的处理.  
完全一样操作的并发测试：基本要求是在同一时刻进行完全一样的操作，这类测试的目的是验证核心模块在  
大量用户使用同一功能时是否正常工作；  
相同/不同功能的子功能并发：每个不同的子功能都模拟一定的用户数量，通过工具来控制并发情况；  
如发送与接收邮件模块的一个测试用例，  
功能：当在线用户达到高峰时，发送和接收普通邮件正常，保证2000个以内用户可以同时访问邮件系统，能够正常发送和接收邮件；  
目的：测试系统2000个以内的用户同时在线时能否正常发送邮件；  
方法：采用LOADRUNNER的录制工具录制一个邮件发送过程测试，要监视数据库服务器和WEB服务器的性能，其中发送的邮件为普通邮件，附件大小不超过1MB.  
并发用户数与事务执行情况：并发用户数，事务平均响应时间，事务最大响应时间，平均每秒处理事务数，事务成功率，每秒点击率，平均流量；  
并发用户数与数据库主机：并发用户数，CPU利用率，MEM利用率，磁盘I/O参数，DB参数；  
并发用户数与应用服务器的关系表：并发用户数，CPU利用率，MEM利用率，磁盘I/O参数；  
4. 组合模块用户并发性能测试的用例设计  
组合模块的性能测试是最能反映用户实际使用情况的测试，它把前面系统中具有耦合关系的模块组合起来进行测试，可以理解为集成性能测试，组合模块并发测试可以真实反映用户使用系统的情况，可以从需求，设计文档；现场调查，系统采集数据获取用户场景；  
具有耦合关系的核心模块进行组合并发测试：主要测试在多用户并发条件下，一些存在耦合关系或者数据接口的模块是否正常运行；  
彼此独立的，内部具有耦合关系的核心模块组的并发测试：这类测试的对象是多个模块组，每个组相关的模块具有一定的耦合关系，组与组之间关系相互独立，主要站在用户的角度考虑问题；  
基于用户场景的并发测试：选择用户的一些典型场景进行测试，测试对象不限制于核心模块或非核心模块；  
组合模块用户并发性能测试的前两种类型仍然是针对核心模块的同时也关注用户场景，这样做的原因是大多数的性能问题都是由用户经常使用的核心模块一起的；可以看出，组合模块的用户并发性能测试既关注功能测试，也关注性能测试，通过发现一些接口和综合性能方面的问题，使系统更加稳定的运行。  
如下某OA系统组合模块的一个测试用例：  
功能：在线用户数达到高峰时，用户可以正常使用系统，目标是满足500个以内用户同时  
在线使用系统；  
目的：测试500个以内用户同时在线时能否使用比较常见的模块：公文系统，电子公告，网上论坛；  
方法：采用LOADRUNNER 的录制工具录制三项业务；业务1，在公文系统内进行打开，修改等操作；业务2，在电子公告系统内，察看发布公告； 业务3 ，在网上论坛系统内发布帖子，查看文章；每项业务分配一定数量的用户，利用LOADRUNNER来完成；  
并发用户数与事务执行情况：业务1，业务2，业务3事务平均响应时间；业务1，业务2，业务3事务最大响应时间；业务1，业务2，业务3平均每秒事务数；业务1，业务2，业务3平均成功率；每秒点击率；平均流量；  
并发用户数与数据库主机：CPU利用率；MEM利用率；磁盘I/O情况；DB参数；  
并发用户数与应用服务器的关系：CPU利用率，MEM利用率；磁盘I/O情况；  
5. 疲劳强度与大数据量测试  
疲劳强度测试：主要特点是长时间对目标测试系统加压，目的是测试系统的稳定性，持续时间一般在1小时以上；疲劳强度测试属于用户并发测试的延续，因此核心内容仍然是核心模块用户并发和组合模块用户并发，在编写测试用例时需要编写不同参数或者负载条件下的多个测试用例，可以参考用户并发性能测试用例的设计内容，通常修改相应的参数就可实现所需要的测试场景；如下疲劳强度测试用例：  
极限名称：200个用户同时使用系统的3个模块；  
前提条件：测试客户端要有足够的资源；  
运行时间：连续运行16小时；  
测试方法：采用LOADRUNNER录制3个任务，然后开始对系统加压；  
输入动作：任务1，任务2，任务3 ；持续时间， 任务20小时， 任务2，21小时，任务3，16小时；用户数量；现象；  
大数据量测试：主要针对对数据库有特殊要求的系统进行的测试，如电信业务系统的手机短信业务；可以分为实时大数据量，主要目的是测试用户较多或者某些业务产生较大数据量时，系统能否稳定运行；极限状态下的测试，测试系统使用一段时间即系统累计一点量的数据时能否正常的运行业务；前面两种的结合，测试系统已经累计了较大数据量时，一些实时产生较大数据量的模块能否稳定工作；如下大数量测试用例：  
功能：数据库中的短信息表可以保存所有不能及时发送的短信息，用户上线后又能及时发送已经保存的信息；  
目的：  
方法：  
并发用户数与事务执行情况：输入说明； 事务平均响应时间；事务最大响应时间；平均每秒处理事务数，事务成功率；每秒点击率；平均流量；  
6. 网络性能测试  
基于硬件的测试：主要是通过各种软件工具，仪器等测试整个系统的网络运行环境，一般由系统集成人员负责；  
基于应用系统的测试：主要测试用户数目与网络带宽的关系，通过测试工具准确展示带宽，延迟，负载和端口的变化是如何影响用户响应时间的；  
网络性能测试的用例设计主要针对后一种类型，可以独立进行测试，也可以和用户并发性能测试，疲劳强度与大数据量测试结合起来，在原有的基础上采用工具来调整网络设置，从而达到监视网络性能的目的；如下网络性能测试用例；  
目的： 测试系统运行在不同网络带宽条件下的性能情况，以及与并发用户数量的关系；  
方法：在不同的广域网带宽下使用LOADRUNNNER录制邮件系统得相关事务操作脚本，  
然后以不同的带宽和并发用户数进行压力测试，并记录在各种用户条件下各种事务的响应情况，同时记录路由器端口的流量和其他数据；  
运行时间：  
并发用户数与事务响应时间：  
7. 服务器性能测试  
服务器性能测试主要是对数据库，WEB服务器，操作系统的测试，目的是通过性能测试找出服务器的瓶颈，为系统扩展，优化提供相关的依据；分为：  
高级服务器性能测试：在特定的硬件条件下，由数据库，WEB服务器，操作系统相应领域的专家进行的性能测试；  
初级服务器性能测试：在系统运行前面的性能测试时，通过测试工具对数据库，WEB服务器，操作系统的使用情况进行监控，然后进行综合分析，找出系统瓶颈；性能测试的主要目的是在软件功能良好的前提下，发现系统瓶颈并解决，而软件和服务器是产生瓶颈的两大来源，因此服务器测试一定要和前面的测试结合起来进行；在进行用户并发性能测试，疲劳强度与大数据量性能测试时，可以完成对服务器的监控并对服务器性能进行评估；这类部分的测试用例一般不必单独编写。  
四、WEB性能测试用例设计  
WEB 性能测试用例设计模型是设计性能测试用例的一个框架，在实际项目中，需要对其进行适当的剪裁，从而确定性能测试用例的范围和类别，裁减的依据是性能测试策略和测试范围；在测试用例主要框架确定后，接下来就要如何设计各类性能测试用例中具体数据。  
基于用户的测试多在用户现场进行，而为了测试目的而进行的测试多在开发环境即开发团队的内部进行；为了测试目的而设计的测试用例场景主要根据测试设计人员的经验来进行，但是仍要参考用户的实际场景，用户实际使用场景是设计所有测试用例的依据，性能测试用例设计首先要分析出用户现实中的典型场景，然后参照典型场景进行设计。比较常见的用户场景有如下三种：一天内不同时段的使用场景；系统运行不同时期的场景；不同业务模式下的场景；各类测试用例设计的细节：  
1. 确定用户使用系统情况的方法  
确定用户对系统的使用情况是设计用例具体数据的基础，后面并发用户数据设计，疲劳强度设计以及各种场景设计都要依赖对用户使用系统情况的分析，分析用户使用情况经常采用现场调查和分析系统日志两种方法；  
用户现场调查：通过和用户进行沟通，可以确定用户的人员组成情况；这类方法适用于用户群体固定且目标测试系统没有投产前的情况；  
分析系统日志：当用户比较分散，现场调查比较困难时，可以采用对系统日志进行分析的方法，作为对用户现场调查的补充；  
2. 并发用户数量设计  
设计并发用户数量前，首先要了解确定系统最大并发用户数量的方法；可以根据系统的最大使用人数或者最大在线数量来评估最大并发用户数量的方法；  
极限法：取最大在线用户数作为最大并发数，这种方法适用于系统已经投产目标用户群体不确定的门户网站，可以通过分析日志来进行测试；也可以使用系统已经注册的用户数量作为系统的用户数量，按照经验公式来估算最大用户数量；  
用户趋势分析：对软件生存周期内的用户未来走势进行分析，预测系统可能达到的最大使用用户数目，从而估算系统的最大并发用户数目，这种方法多用于用户数目逐渐增多的情况；  
经验评估法：多用于系统的使用用户数目相对稳定而且比较明确的系统；  
并发用户数量的设计基本是按照最大并发用户的数量的百分比来设计的，对于某一特定的用例，需要注意：  
一按照各类用户同时递增的方式来设计用户数量，是为了按照由浅入深的方法来发现系统的瓶颈；二并发用户的最大值一般不会超过前面计算的最大并发用户数量的20% ，除非是为了测试系统能支持的最大并发用户数量；三设计用户数量时要考虑成本，因为每组用户数都意味着至少执行一次测试；  
3. 系统不同时间段场景的设计  
不同时间段的场景更接近用户使用情况，它也是设计核心模块和组合模块并发性能测试用例的基础，不同时间段场景分析的数据主要是前面的需求分析和日志分析结果；不同时间段场景的设计基本原则有两个：一是选择典型的场景进行测试；尤其要选择场景中并发用户数目较大的场景；二是要覆盖全面，设计出的用例要覆盖到压力可能较大的时间段；用户场景的设计一般与后面的业务模式结合起来进行；  
4. 业务模式的设计  
业务模式的设计是不同时间段场景设计的特例，也是设计核心模块和组合模块并发性能测试用例的基础，设计业务模式的目的是专注于某些功能模块的组合，按时间段来设计场景通常会涉及很多模块，如果系统存在的由应用软件引起的瓶颈则很难定位，所以才抽象一些特定的业务模式来进行用例的设计；  
按照业务模式和时间段的场景来设计性能测试用例时，会涉及到如何设计每个模块并发用户数目的问题，通常会取各个相关模块在24小时内最大的并发用户数目进行组合；  
5. 大数据量测试用例的设计  
历史数据相关的大数据量测试设计与并发用户的测试设计很类似，首先要确定系统数据的最长迁移周期，确定了系统的最大数据量后，接下来选择一些前面的核心模块或者组合模块的并发用户测试用例作为其主要内容即可；  
运行时大数据量测试主要根据模拟系统运行时可能产生的大数据量来进行测试，这类测试用例通常根据实际情况去分析设计；  
6. 一些特定测试用例的设计  
疲劳强度测试，最大用户测试，容量测试等一些特殊的测试用例设计，根据用户的需求进行，这类用例的相关要求通常十分明确；  
性能测试用例最重要的是注意用例间的关系，孤立的设计各类用例只能增加测试成本，浪费人力。性能测试用例设计人员应该追求设计既能覆盖性能测试需求，又能以较低的成本来执行测试用例；  
五、WEB性能测试用例设计总结  
1. 测试用例可用性总结  
对于一个比较完善的性能测试项目，经常会有一些测试用例不能执行，，因此测试完成后应该分析哪些用例不能执行以及不能执行的原因，这样可以为下次测试打好基础。  
2. 用例执行效果分析  
通过对用例执行效果进行分析，可以为升级或者开发新的性能测试用例提供有利的参考，不是所有的用例都能导致系统瓶颈的出现，因此应该分析哪些用例能够发现系统问题，哪些用例执行时没有太大效果。分析那些设计好的用例不但有助于以后设计用例，还可以为再次执行提供参考：当下次测试进度压力较大时可以先执行重要的用例，跳过那些尝试性的，不容易发现问题的用例；  
3. 用例执行时间分析  
分析用例的执行时间是为下次规划性能测试提供参考，由于很多用例执行时间不是特别确定，导致性能测试计划也具有一定的不确定性。通过分析用例的执行时间可以为以后的制定测试计划提供参考；  
总之，性能测试用例的设计是需要通过不断分析总结才能做好，不但要分析性能测试用例的  
可用性，执行效果，执行时间，还应该分析用例的设计方法，设计思路等。