性能测试

2019年11月11日

0:53

用户视角

响应时间

管理员视角

响应时间

系统状态相关信息

开发的视角

响应时间

扩展性

性能瓶颈

术语解释：

资源：指在服务端存放的数据，文字，图片，多媒体文件等。

响应时间

呈现时间：数据在被客户端收到响应数据后呈现页面所消耗的时间

系统响应时间：应用系统从请求发出开始到客户端接收到数据所消耗的时间

并发用户数

计算公式：

平均并发用户数 C=n\*L/T

并发用户数峰值CC=C+3\*sqrt(C)

吞吐量

单位时间内系统处理客户请求的数量

吞吐量一般用请求数/秒或页面数/秒来衡量

在web系统的性能测试过程中，吞吐量主要以请求数（单击数）/秒，页面数/秒或者字节数/秒来体现。

计算方法：

吞吐量F=Nvu\*R/T

F吞吐量

Nvu虚拟用户数量

R表每个vu发出的请求数量

T表性能测试所用的时间

性能计数器

描述服务器或操作系统性能的一些数据指标

计数器在性能测试中发挥着”监控和分析“的关键作用

思考时间

用户在进行操作时，每个请求的间隔时间

0思考时间

三个环境

环境（硬件组成，网络类型，连接方式）

开发环境，测试环境，生产环境

生产环境：指系统上线后，实际运行的环境。

测试环境：

开发环境：

性能测试的方法

性能测试：

通过模拟生产环境的业务压力和使用场景组合，测试系统的性能是否满足实际的性能的需求

主要目的验证系统是否有系统宣称具有的能力

需要事先了解被测系统典型场景，并具有确定的性能目标

负载测试

在一定负载成都下进行最大负载的过程

主要目的是找到系统处理能力的极限

需要在给定的测试环境下进行，通过逐步施加压力的方式找到系统性能的最大瓶颈

用来了解系统的性能容量，或是配合性能调优来使用

压力测试

通过确定一个系统的瓶颈或者不能接受性能点，来获取系统能提供的最大服务级别的测试

主要目的是为了价差系统处于压力情况下，应用的表现

主要通过模拟负载等方法，是的系统的资源使用较高水平（80%~90%）

一般用于测试系统的稳定性

负载测试和压力测试的区别：

负载测试是逐级施压，压力测试是恒定。

配置测试

对被测系统的软硬件环境的调整，了解各种不同的环境对系统性能的影响的程度，从而找到系统各项资源的最优配置

主要目的是为了了解各种不同因素对系统性能的影响的程度，从而判断出最值得进行调优操作。

一般在对系统性能状况有初步了解后进行

一般用于性能调优或规划能力。

并发测试

通过模拟用户，得并发访问，测试多用户并发访问同一个模块或子系统时是否存在死锁或者其他性能问题

主要目的是发现系统中可能隐藏的并发访问时的问题

主要关注系统可能存在的并发问题

可靠性测试

通过给系统加载一定的业务压力（如资源在60%~80%的使用率）的情况下，让应用持续运行一段时间，测试系统在这种情况下是否能够稳定运行

主要目的时检验系统是否长期稳定的运行

需要在压力下持续一段时间运行

测试过程中需要关注系统的运行状况

压力测试和可靠性测试的区别：

资源使用率的不同，压力测试资源使用率在80-90，可靠性测试资源在60-80.

结束的判断条件不同，压力测试在系统奔溃后停止，可靠性在系统出错后停止。

系统崩溃和系统出错是俩个量级的事情，系统出错到系统崩溃中间还可能会经历一个很长的时间

失效恢复测试

针对有冗余备份和负载均衡的系统设计的。这种测试方法可以用来检验如果系统局部发生故障，用户是否能够继续使用系统；以及如果这种情况发生，用户将受到多大程度的影响

主要目的是验证在局部故障的情况下，系统能否继续使用。

当问题发生时能支持多少用户访问和采取何种应急措施

只有对系统持续运行指标有明确要求的系统才需要进行这种类型的测试。

全面性能测试模型

预期指标的性能测试

需求分析与设计阶段的性能指标

独立业务性能测试

组合业务性能测试

疲劳强度性能测试

长时间处理大业务量的能力

大数据量性能测试

在生产环境下进行

网络性能测试

一般由系统集成人员完成

服务器性能测试

特殊测试

配置测试，内存泄漏测试

性能调优

协助解决问题的过程就叫性能调优

性能调优的步骤

确定问题：（顺序）

检查代码

调整数据库配置

检查硬件配置

检查软件配置

检查网络

确定原因：

影响

用户数

资源监控

模块

客户端/服务端

硬件配置

负载

优化

确定调整目标和解决方案

确定调整目标的主要作用是明确何时停止系统调整，否则工作将永无尽头

测试解决方案

分析调整结果

系统调整是否达到或超出了预定目标

系统是否整体性能得到了改善，还是牺牲了某部分性能来解决问题

调整是否可以结束了。

如何做好性能测试

规范的管理流程

开发阶段的性能测试

用户现场的性能测试

需求分析—规划与设计—执行—调优—验证

提高测试人员在性能测试方面的技能

“七种武器”（硬件，网络，数据库，编程，使用工具，专业能力数据分析，业务知识）

行业业务知识

loadrunner入门

loadrunner作为专业的性能测试工具，通过模拟成千上万的用户对被测应用进行操作和请求，在实验室环境中精确重现生产环境中任意可能出现的业务压力，然后通过在测试过程中获取信息和数据来确认和查找软件的性能问题，分析性能瓶颈

loadRunner的测试流程（场景设计优先于脚本录制）

制定测试计划

创建测试脚本

创建场景

运行场景

监控测试场景

分析测试结果

LoadRunner的三大模块

Virtual user Generator —录制脚本

LoadRunner Controller —创建，运行和监控场景

LoadRunner Analysis —分析测试结果

制定压力/负载测试计划

分析被测系统

预计有多少用户会连到系统

客户机的配置情况（spu，内存，操作系统，软件工具等）

服务器使用什么类型的数据库以及服务器的配置情况

客户机和服务器之间如何通讯

还有什么组件会影响Response Time 指标（如 Modem等）

通讯装置（网卡，路由器）的吞吐量，每个通讯装置能够处理多少并发用户

确定测试目标

系统完成某个事务所花费的时间

明确何种配置能够使系统性能最优

系统能够无差错运行的最大时间

设计在loadrunner中怎样执行

使用Virtual User Generator创建脚本

要创建一个自动测试，首先要创建自动测试脚本，测试脚本的录制和维护是负载测试的重要步骤之一，成千上万的虚拟用户正是通过执行测试脚本来对系统施压的。使用Loadrunner的virtual User Generator 引擎，可以监视并记录客户端和服务器之间的通话，让虚拟用户模拟实际的业务流程，记录真正用户的操作行为，并将其转化为特定的测试脚本语言集合

使用Controller创建测试场景

创建场景的目的是为了实现真实的负载，也就是让一台或多台机器模拟多个用户，同时执行脚本，对被测应用进行操作或发起请求

每个LoadRunner的场景都相当于一套负载测试方案，因为它包括了反应业务操作流程的测试脚本，反映系统承受能力的虚拟用户数量，用户平衡测试机自身压力的Load Generator 机器及脚本之间的执行顺序等，执行场景的过程就是对系统施压的过程。

实时监控场景

LoadRunner集成了实时的监控器，在场景执行过程中，用户可以根据需要选择一个或多个监视窗口对关心的数据进行动态监控，可监控的信息包括系统资源，网络设备，web服务器和数据库等交易数据。

LoadRunner还提供ContentCheck TM工具帮助用户从终端角度观察程序性能情况，判断负载下的应用程序的功能是否正常

分析测试结果

LoadRunner提供分析和报告工具——Mercury LoadRunner Analysis,能够打开这些结果数据，帮助用户找到性能问题并追溯原由