

软件性能测试概述

杨金凤, 孟岩

(国网电科院 软件质量评测中心, 安徽 合肥 230000)

摘要: 随着对应用系统本身性能的关注, 性能测试作为检测系统性能能否满足需求的重要手段应运而生。该文概述了性能测试应用领域和测试方法, 并对测试流程和测试过程中的监控工具和性能计数器进行了简介, 最后通过一个例子说明性能测试的具体操作。

关键词: 性能测试方法; 性能计数器; 性能测试案例

中图分类号: TP311 **文献标识码:** A **文章编号:** 1009-3044(2011)34-8886-03

Software Performance Testing Overview

YANG Jin-feng, MENG Yan

(Software Testing Center of State Grid Electric Power Research Institute, Hefei 230039, China)

Abstract: With the close attention was paid on performance of application, software performance testing come into being, as a safeguard for the dection of system performance. What fields, methods, tools and counters of performance testing was briefly introduced. It was illustrated that specific operation of performance testing by an example.

Key words: performance testing method; performance counter; performance testing case

目前, 信息业务系统已出现信息用户大量增长的态势, 系统的性能问题也逐步成为了我们关注的焦点, 系统能稳定在什么样的性能水平, 或者面临公司业务冲刺时, 系统能否经受住“考验”等这些问题, 都需要通过一个完整的性能测试来给出答案。

性能测试主要是获得模拟真实用户环境, 对系统状况和性能进行预测。

1 性能测试简介^[1]

对于要上线的系统, 如何确定它的性能可以满足用户的需求; 当系统的用户数将大幅增加时, 应当如何调整系统, 是增加应用服务器还是提高数据库服务器的配置, 哪种方法才是最有效的。这些问题都和软件性能、软件性能测试相关。但不同的问题, 在测试中关注的内容会明显的不同, 为了了解这些不同的关注内容, 在测试中肯定也应当采用不同的性能测试方法。

1.1 应用领域

性能测试的应用领域可以化为4个组成部分: 能力验证、规划、调优、发现缺陷。

一个典型的能力验证问题会采用这样的描述方式: “某系统能否在A条件下具有B能力。”能力验证要求在已确定的环境下进行测试, 在设计测试方案和用例时应根据典型场景。

“某系统能否支持未来一年内的用户的一倍增长”或是“应该如何调整系统配置, 使系统能够满足增长的用户数的需要”的问题描述, 通常是规划能力应用领域内的问题, 是探索性测试。

性能调优是即为字面意思。需要注意的是保住每次执行时数据库具有相同的数据环境。而且, 需要具有一个可用于比较的测试基准测试环境。

发现缺陷主要是采用并发测试的方法来发现系统可能存在的问题。

1.2 性能测试的方法^[2]

性能测试(狭义)和负载测试是常见的类型, 此外, 还有压力测试、配置测试、并发测试、可靠性测试、失效恢复测试。

性能测试(Performance Testing)(狭义): 在已确定的环境下运行典型的场景, 验证系统系统是否达到预期的性能目标。

负载测试(Load Testing): 不断增加压力, 直到某个性能指标或者某种资源使用已经达到饱和状态, 也称为可量性测试。目的是找到系统处理能力的极限。

压力测试(Stress Testing): 测试系统在一定饱和状态下, 如CPU、内存等在饱和使用情况下, 系统能够处理的会话能力, 以及系统是否会出现错误。出于这样的考虑: “如果一个系统能够在压力环境下稳定运行一段时间, 那么这个系统在通常的运行条件下应该可以达到令人满意的稳定程度”, 通常可以使用此方法判断系统的稳定性。

配置测试(Configuration Testing): 对被测系统的软/硬件环境的不断调整, 找到系统各项资源的最优分配原则。一般用于性能调优和能力规划。

收稿日期: 2011-10-10

作者简介: 杨金凤(1984-), 女, 安徽大学计算机专业硕士研究生, 从事软件测试工作; 孟岩(1986-), 女, 毕业于合肥工业大学计算机专业, 从事软件测试工作。

并发测试(Concurrency Testing):通过模拟多个用户的并发访问,测试模块或者数据记录是否存在并发问题,例如内存泄露、线程锁和资源争用等。

可靠性测试(Reliability Testing):通过给系统加载一定的业务压力(例如资源在70%~90%的使用率)的情况下,持续运行一段时间,检查系统是否能稳定运行。通常,对一般的非关键的大型应用来说,一般让系统在峰值压力下,运行2~3天基本上就已经够了。

失效恢复测试(Failover Testing):是针对有冗余备份和负载均衡的系统设计的。

2 流程概述

性能测试通常由五个阶段组成:测试计划、脚本创建、场景定义、场景运行和结果分析,如图1所示。如果是划成六个阶段,则包含测试报告部分。

2.1 制定测试计划

了解该系统的功能,制定测试任务的优先级,预测负载的最高峰出现的情况。接着,确定测试的目标,例如“某个事物处理需要的时间”,“哪种配置能够提供最佳的性能”或者“系统扩充后,性能或可靠性是否受影响”等。最后,确定需要度量哪些性能参数,例如CPU使用率、典型业务的响应时间等。

2.2 创建脚本

以工具LoadRunner为例,首先将最终用户活动捕获到Vuser自动脚本中,录制基本的用户脚本。然后通过插入事物和集合点等方法完善测试脚本,以增强脚本的灵活性。

2.3 场景定义

模拟大量用户的操作,通过配置和执行场景向服务器产生负载,验证系统的各项性能是否到达用户的要求。

2.4 测试执行

在测试执行,需要先对单个脚本单用户执行,以获得基准数据,然后在按照设定好的场景运行。在场景执行过程中,需要监视各个服务器的运行情况(具体性能计数器参见3.1性能计数器),例如响应时间、吞吐量、资源利用率等,确定系统瓶颈。

2.5 结果及分析

分析结果了解系统的性能状况并能够对性能进行提高。影响终端用户响应时间的瓶颈包括应用程序和服务器的吞吐量、终端到终端的Internet连接速度以及网络堵塞等,在运行过程中,可以监视各个服务器的运行情况。

3 性能计数器与工具介绍

3.1 性能计数器

监控指标根据性能需求确定。表1给出的是常使用监控的性能计数器。

3.2 工具介绍^[3]

可以将性能测试的工具划分为三类:负载加压机、资源监控类、故障诊断类。有很多工具集成了这三类功能,例如,普遍使用的LoadRunner。

主流负载性能测试工具:QA Load、SilkPerformer、LoadRunner以及开源的OpenSTA等。最为普遍使用的是LoadRunner,它通过模拟实际用户的操作行为和实行实时性能监测,更快的确认和查找问题。

资源监控工具:UNIX主机用户可以直接使用topas,vmstat,iostat了解系统自身的健康工作状态;weblogic以及websphere平台可以通过自身的监控台,在上面可以了解到目前的JVM的大小、数据库连接池的使用情况和目前连接的客户端数量及请求状况等。目前的绝大多数的监控工具基本上是直接从中间件、数据库以及主机自身提供的性能数据采集接口获取性能指标,依赖于被监控平台自身的数据采集能力。

故障定位工具以及调优工具:在目前的主流测试工具

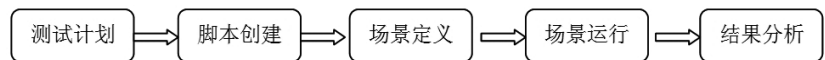


图1 性能测试五个阶段

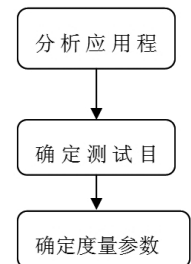


图2 制定测试计划

表1 常监控的性能计数器

监控对象	性能计数器	描述
CPU	%Processor Time	如果该值持续超过90%,表明是瓶颈。
	%User Time	应用程序消耗的CPU时间比。
	%DPC Time	CPU消耗在网络处理上的时间,此值越低越好。
Process	Private Bytes	此进程所分配的无法与其它进程共享的当前字节数量。如果系统性能随着时间而降低,则此计数器可以是内存泄露的最佳指示器。
Physical Disk	Average Disk Queue Length	指读取和写入请求的平均数。该值应不超过磁盘数的1.5~2倍。
	Average Disk sec/Read (Transfer)	指以秒计算的在此磁盘上读取(写入)数据的所需平均时间。
内存	Available Mbytes	可用物理内存数
	Swap(KB)(Unix/Linux系统)	以使用的虚拟内存数量。
	Pages Read/sec	阈值为5,数值越低越好。大数值表示是磁盘读而不是缓存读。
	Cache Bytes	文件系统缓存(File System Cache),默认情况下为50%的可用物理内存。
数据库	Cache Hit Ratio	cache的命中率。当该值比较小,而数据库比较繁忙时,可能需要调整缓存的大小。
	PGA Memory/UGA Memory (仅用于Oracle)	Oracle数据库进程的内存情况。
	Total ServerMemory(仅用于SQL Server)	SQL Server数据库进程当前使用的内存量。
	User Connections	当前的用户连接数。数据库服务器一般都有用户连接数的限制,当应用不合理时,有可能出现连接数超过限制的情况,导致一些异常的发生。
应用服务器	JVM Heap Size	JVM堆大小,该计数器的值是个实时值。
	JVM Heap Free	JVM可用堆大小,该计数器的值是个实时值。
	Pending Request Oldest Time (Execute Queue)	队列请求的最久时间。可以看到队列有无明显的拥塞情况。

厂商中,都相应地提供了对应的产品支持。比如Loadrunner模块中添加的诊断以及调优模块,Oracle自身提供的强大的诊断模块。

4 实例解析

某网站稿件管理发布系统,在上线之前,需要确定该系统性能是否达到以下需求:1)可以连续稳定运行12小时;2)CPU的使用率要低于80%,内存使用率低于75%;3)查询响应时间要低于2秒,上传稿件的时间要低于5秒;4)能够提供的最大事务处理能力为5笔/秒;5)能够支持最大50个用户并发。

分析:易知,测试性能测试做的是能力验证。需求1)是对系统的可靠性要求,所以需要使用可靠性测试方法;需求2)、4)、5)需要采用负载测试;需求3)在做负载测试和可靠性测试的过程即可验证。

以采用LoadRunner工具为例,具体介绍测试过程^[4]。

1)使用VuGen分别录制查询和上传稿件的脚本,并可以采用以下方法对录制的脚本进行优化

① 插入事务,运行到该事务的开始点时,LoadRunner就会开始计时,直到运行到该事务的结束点,计时结束。通过插入事务点,可以清晰的了解到每个事务操作的响应时间,以确定发生瓶颈的语句。

② 插入集合点,插入集合点是为了衡量在加重负载的情况下服务器的性能情况。

③ 参数化脚本,为了更加真实的模拟实际环境,需要各种各样的输入,此时需要对脚本进行参数化。插入函数,VuGen中可以使用C语言中比较标准的函数和数据类型。

④ 插入Text/Image检查点,检查网页上是否存在指定的Text或者Image,验证Web服务器返回的网页是否正确。

2)根据实际使用情况在Controller中定义场景。(假设平时30%的用户使用的是上传稿件功能,70%使用的是查询稿件功能。)

选择Goal-Oriented Scenario为负载测试的场景类型,将“Transactions per Second”设为预定目标,值设定为5,30%的用户分配给上传稿件脚本,70%的用户分配给查询稿件脚本。选择Manual Scenario为可靠性测试的场景类型,设定15个用户执行上传稿件脚本,35个用户执行查询稿件脚本,运行时间设定为12小时。

3)首先单用户运行脚本,获得基准数据,然后按照设定的场景运行脚本,监控的CPU和内存的使用情况,以及随着用户的增加,运行时间的持续事务的响应时间。

4)Analysis分析结果

Transaction Response Time图。可以判断每个事务完成用的时间,从而可以判断出哪个事务用的时间最长,哪些事务用的时间超出预定的可接收时间。

Windows Resource图,实时地显示了Web Server系统资源的使用情况,可以把瓶颈定位到特定机器的某个部件上。

验证响应时间、CPU和内存的使用率是否符合需求,验证最大能够支持的并发用户数是否为50个。

5 结束语

在性能测试应用领域里,能力验证与规划是性能测试应用领域中相对容易的部分,如何对系统进行调优是更高阶段的技术。

参考文献:

- [1] 柳纯录.软件评测师教程[M].北京:清华大学出版社,2005.
- [2] 薛冲冲,陈坚.软件测试研究[J].计算机系统应用,2010,20(2).
- [3] 徐进.自动化软件测试的分析[J].信息技术,2010(3).
- [4] 于涌.精通软件性能测试与LoadRunner实战[M].北京:人民邮电出版社,2010.