性能测试

CNIC R&D 质量组

议程

- 负载压力测试概述
- 负载压力测试解决方案
- 负载压力测试指标
- 负载压力测试实施
- 负载压力测试技巧
- 实例讨论和演示

负载压力测试概念

- 负载压力测试是指在一定约束条件下测试系统所能承受的并发用户量、运行时间、数据量，以确定系统所能承受的最大负载压力。
- 负载压力测试有助于确认被测系统是否能够支持性能需求，以及预期的负载增长。
- 负载压力测试不只是关注不同负载场景下的响应时间等指标，它也要通过测试来发现在不同负载场景下会出现的例如速度变慢、内存泄漏等问题。
- 应该在开发过程中尽可能早地进行负载压力测试。

负载压力测试目的

1. 在真实环境下检测系统性能，评估系统性能以及服务等级的满足情况。
   - 比如电信计费软件,众所周知,每月二十日左右是市话交费的高峰期,全市几千个收费网点同时启动。如此众多的交易同时发生,对应用程序本身、操作系统、数据库服务器、中间件服务器、网络设备的承受力都是一个严峻的考验。
   - 决策者需要模拟系统负载压力，预见软件的并发承受力，这是在测试阶段就应该解决的重要问题。
   - 这里我们强调真实环境下检测系统性能，在实施过程中大家认为这样做会遇到很多阻力。那么在条件不允许的情况下，我们可以使用一种“模拟环境”来做测试，这种环境指与实际真实应用环境基本等级保持一致的测试环境。

负载压力测试目的

2. 预见系统负载压力承受力，在应用实际部署之前，评估系统性能。
   - 检测系统性能强调对系统当前性能的评估。通过评估，可以在应用实际部署之前，预见系统负载压力承受力。
   - 这种测试的意义在于指导系统总体设计，既可以避免浪费不必要的人力、物力和财力，又避免硬件和软件的设计不匹配，使系统具有更长、更健壮的生命力。
   - 对于系统性能检测，有时我们所从事的工作是仅仅是被动监控一些性能指标，而预见系统负载压力承受力则不可避免会借助自动化的负载压力测试工具。

负载压力测试目的

3. 分析系统瓶颈、优化系统。
   - 瓶颈这个术语来源于玻璃瓶与瓶身相比收缩了的部分。收缩的瓶颈将引起流量的下降，从而限制了液体流出瓶外的速度。类似的，在负载压力测试中，“瓶颈”这个术语用来描述那些限制系统负载压力性能的因素，即应用系统中导致系统性能大幅下降的原因。
   - 瓶颈可能定位在硬件中，也可能定位在软件中。硬件瓶颈与软件瓶颈相比，我们更建议先解决软件瓶颈。
   - 优化调整系统是发现瓶颈，故障定位之后要完成的事情，实现优化之后即可消除瓶颈，提高性能。
   - 导致系统性能下降的因素来自许多方面，例如I/O过载、内存不足、数据库资源匮乏、网络速度低、应用程序架构存在缺陷，软硬件配置不恰当等等。比如是磁盘I/O导致了系统瓶颈，那么消除它的方法可能是重新设计数据库或者改进数据库访问方式。

负载压力测试策略

- 负载压力测试可以采取利用手工进行测试和利用自动化测试工具进行测试两种测试策略。
- 手工模拟负载压力，方法是找若干台电脑和同样数目的操作人员在同一时刻进行操作,然后拿秒表记录下响应时间，这样的手工测试方法可以大致反映系统所能承受的负载压力情况。
- 利用自动化负载压力测试工具进行测试可以在一台或几台PC机上模拟成百或上千的虚拟用户同时执行业务的情景，通过可重复的、真实的测试能够彻底地度量应用的性能，确定问题所在。
- 利用商业化的自动化测试工具是进行负载压力测试的主要手段，知名的商业化的测试工具比如LoadRunner、QALoad等适应范围非常广，一般都经过了长时间的市场检验，测试效果得到业界的普遍认可，测试结果具有一定的可比性。

负载压力测试解决方案

系统的负载压力测试主要包括并发性能测试、疲劳强度测试以及大数据量测试。

- 并发性能测试

  并发性能是负载压力性能的最主要的组成部分。对一个系统来讲，某些业务操作对特定角色用户来讲存在很大的同时操作的可能性，并发性能的测试对于保证系统的性能是非常关键的。

- 疲劳强度测试

  疲劳强度对系统来讲也是一种负载，它强调的是长时间的考核。

- 大数据量测试

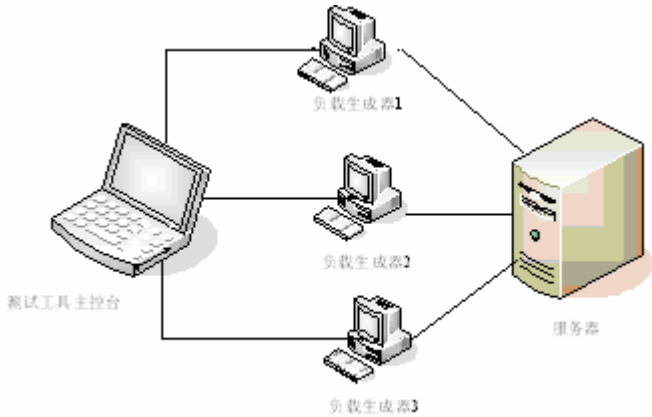
  大数据量测试包括独立数据量测试和综合数据量测试两种主要类型。

并发性能测试

- 并发负载压力的实施是在客户端，负载压力的传输介质是网络，最终压力会到达后台各类服务器。
- 在并发性能测试过程中，需要关注点：
  - 应用在客户端的性能
  - 应用在网络上的性能
  - 应用在服务器上性能
- 测试要定位问题所在，目的是为了解决问题，这些关注点正是定位问题的必要条件。

应用在客户端性能的测试

- 在客户端模拟大量并发用户执行不同业务操作，达到实施负载压力的目的。
- 采用负载压力测试工具来模拟大量并发用户，主要组成部分包括主控台、代理机以及被测服务器，各部分采用网络连接。
- 主控台负责管理各个代理以及收集各代理测试数据，代理负责模拟虚拟用户加压。在每次并发性能测试中，只有一台主控台，但可以有多个代理。

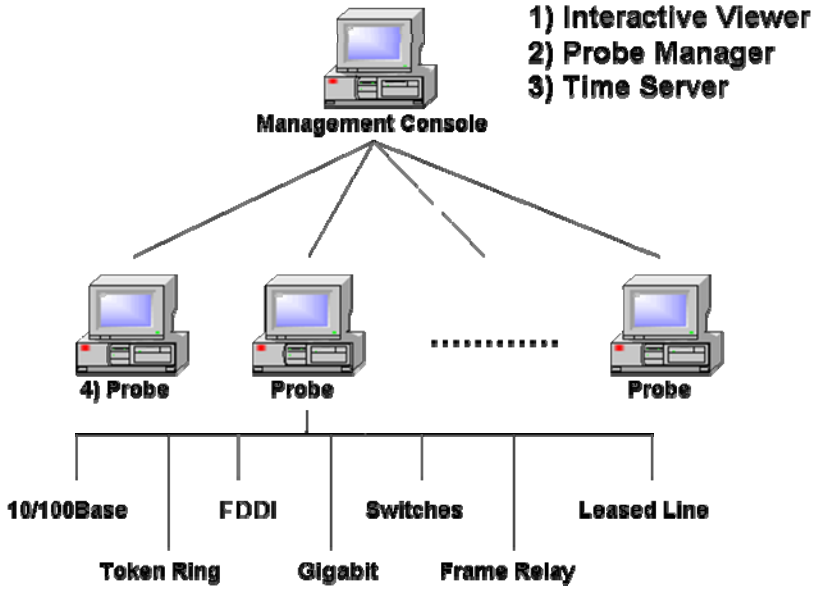


应用在客户端性能的测试

- 如何模拟负载压力？总的原则是最大限度地模拟真实负载压力。要做到这一点，既是对测试工具的考验，又是对测试工程师经验与智慧的考验。
- 要模拟真实的负载压力做测试，必须创建方案，方案是用以模拟现实生活中的用户的方式。方案包含有关如何模拟实际用户的信息：虚拟用户 (Vuser) 组、Vuser 将运行的测试脚本、多个测试脚本的相互关联、Vuser 运行脚本的策略，以及如何使用负载生成器产生并发压力等。
- 客户端并发性能测试的指标值，以及怎样分析结果值，我们在稍后会有详细论述。

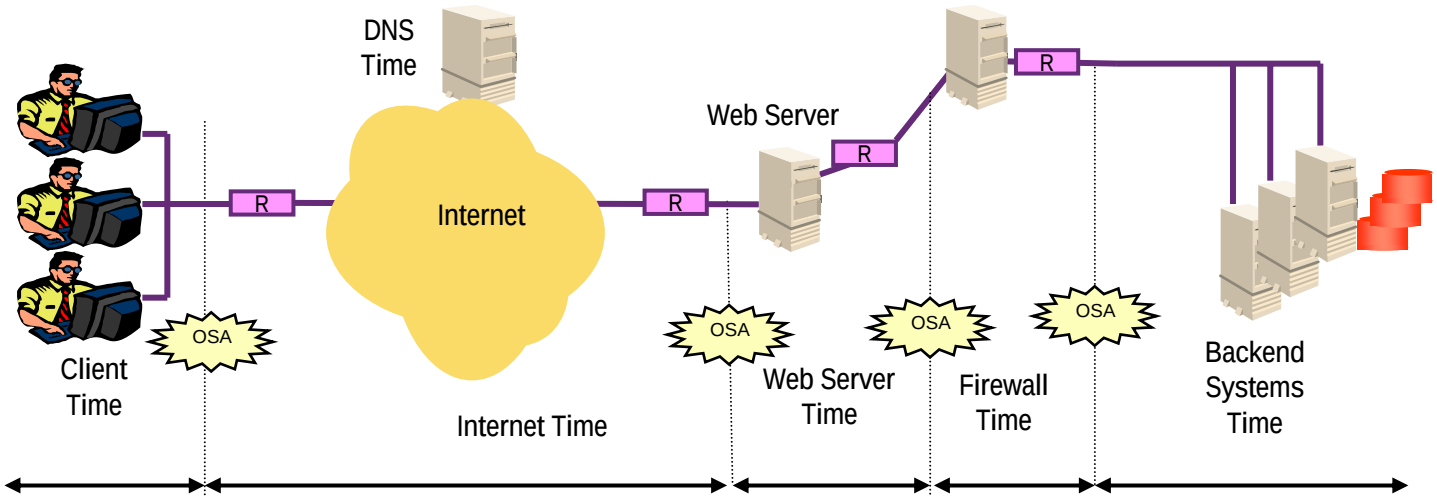
应用在网络上性能的测试

- 应用在网络上性能的测试包括两部分内容，一是应用网络故障分析；二是网络应用性能监控。
- 应用网络故障分析的测试目标是显示网络带宽、延迟、负载和TCP端口的变化是如何影响用户的响应时间的。其工作机理可以总结为“多个捕捉点，一个分析”。捕捉点即“Agent”，利用主控台“Agent Manager”进行分析，Agent被动监听数据包来实现实时数据采集，Agent Manager完成对所跟踪到的数据的分析。



应用在网络上性能的测试

- 下图是一个典型的Web架构应用部署图，我们可以在应用逻辑路径上多点数据采集，并且在任何两个节点间进行数据整合，测量分段的响应时间，分析应用故障。



应用在网络上性能的测试

- 网络应用性能监控的测试目标是在系统试运行之后，我们需要及时准确地了解网络上正在发生什么事情；什么应用在运行，如何运行；多少PC正在访问LAN或WAN；哪些应用程序导致系统瓶颈或资源竞争。
- 利用工具进行网络应用性能监控，监控探针可以部署在整个应用网络上。监控工具主要包括以下部分：
  - 探针

    采集与存储数据，并根据应用对数据进行分类。设置的原则是根据网络组成和监控要求。
  - 探针管理器

    管理配置探针，设定数据采集与上传时间，合并收集的数据。
  - 时间服务器

    对探针进行时针同步。
  - 交互界面

    数据展示平台。

应用在服务器上性能的测试

- 这里我们谈到的是“测试”的概念就是对服务器执行监控，监控的内容主要包括操作系统、数据库以及中间件等内容。目前监控的手段可以采用工具自动监控，也可以使用操作系统、数据库、中间件本身提供的监控工具。利用工具监控有下列优点：
  - 减少故障诊断和分析时间；
  - 减少手工定位的时间和避免误诊；
  - 在问题发生前定位故障；
  - 验证可达到的性能水平和服务水平协议；
  - 持续的服务器、数据库和应用性能和可用性监控；
  - 服务器、应用可用性和性能报告。
- 操作系统、数据库、中间件本身提供的监控工具有时采用命令行的方式，有时具备友好的图形界面，例如监控UNIX服务器资源占用可以使用vmstat，或者iostat命令，Web应用中间件Websphere的监控可以采用系统本身提供的Web页面的监控工具。

疲劳强度测试

- 疲劳强度对系统来讲也是一种负载，它强调的是长时间的考核。
  - 日常业务疲劳强度模拟

    日常业务疲劳强度测试就是模拟系统的日常业务，持续执行“一段时间”，暴露系统的性能问题，例如内存泄漏、资源争用等，分析与调整的方法与并发性能测试是非常类似的。
  - 高峰业务疲劳强度模拟

    一般情况下系统运行都有其高峰期，比如对一个鲜花定购系统而言，在特殊的节日，例如情人节、母亲节等都是其高峰期，疲劳强度测试必须要模拟这样的高峰业务。

    这里我们需要对“一段时间”有个合理的选择，这个时间指标要满足两个主要条件，一是这段模拟时间所处理的交易量要达到系统疲劳强度需求的业务量，二是在这段测试周期中必须通过加大负载，以及尽可能长的测试周期来保证疲劳强度测试。

大数据量测试

- 大数据量测试包括独立数据量测试和综合数据量测试两种主要类型。
  - 独立数据量测试

    针对某些系统存储、传输、统计、查询等业务进行单用户大数据量测试。例如对某些系统经常会有上传、下载的操作，操作的对象可能就是大数据量，包括图片文件、音频文件或者视频文件等等。还有些系统存在大量的批处理任务。
  - 综合数据量测试

    我们提出“一定的数据量是并发测试与疲劳测试的基础”，在并发测试和疲劳强度测试过程中如果不考虑数据量对系统性能的影响，无疑就带来一个缺陷。
  - 测试数据的生成

    利用自动化负载压力测试工具，模拟用户业务操作，同时并发数百个或者数千个用户生成相关数据，这样测试工程师并不需要清楚地知道数据表与表之间的关系等细节内容，就可以制造大量的测试数据。

负载压力测试指标

我们可以选择的指标包括以下几类:

- 客户端交易处理性能指标;
- 服务器资源监控指标，例如: UNIX;
- 数据库资源监控指标，例如: Oracle;
- Web服务器监控指标，例如: Apache;
- 中间件监控指标，例如: Websphere等等。

这里要说的是: 指标的选择并非越多越好。选取什么样的指标要以测试需求为准，想得到什么类型的性能值就选择相应的指标。过多的无用指标会大量占用各种资源，并对测试结果产生影响。

交易处理性能指标

- 并发用户数
- 平均事务响应时间
- 每秒事务数
- 每秒事务总数
- 每秒点击次数
- 吞吐量
- HTTP 状态代码摘要
- 每秒HTTP 响应数
- 每秒下载页面数

UNIX操作系统资源监控

度量	描述
Average load	上一分钟同时处于“就位”状态的平均进程数
Collision rate	每秒钟在以太网上检测到的冲突数
Context switches rate	每秒钟在进程或线程之间的切换次数
CPU utilization	CPU的使用时间百分比
Disk rate	磁盘传输速率
Incoming packets error rate	接收以太网数据包时每秒钟接收到的错误数
Incoming packets rate	每秒钟传入的以太网数据包数
Interrupt rate	每秒内的设备中断数
Outgoing packets errors rate	发送以太网数据包时每秒钟发送的错误数
Outgoing packets rate	每秒钟传出的以太网数据包数
Page-in rate	每秒钟读入到物理内存中的页数
Page-out rate	每秒钟写入页面文件和从物理内存中删除的页数
Paging rate	每秒钟读入物理内存或写入页面文件中的页数
Swap-in rate	正在交换的进程数
Swap-out rate	正在交换的进程数
System mode CPU utilization	在系统模式下使用CPU的时间百分比
User mode CPU utilization	在用户模式下使用CPU的时间百分比

Windows操作系统资源监控

对象	度量	描述
System	% Total Processor Time	系统上所有处理器都忙于执行非空闲线程的平均时间的百分比。
Processor	% Processor Time (Windows 2000)	处理器执行非空闲线程的时间百分比，此计数器设计为处理器活动的一个主要指示器。
System	Processor Queue Length	线程单元中的处理器队列的即时长度。此值为即时计数，不是一段时间的平均值。
Memory	Page Faults/sec	此值为处理器中的页面错误的计数。
PhysicalDisk	% Disk Time	选定的磁盘驱动器对读写请求提供服务的已用时间所占百分比
Memory	Pages/sec	为解析内存对页面（引用时不在内存中）的引用而从磁盘读取的页数或写入磁盘的页数。这是“Pages Input/sec”和“Pages Output/sec”的和。
System	Total Interrupts/sec	计算机接收并处理硬件中断的速度。
Objects	Threads	计算机在收集数据时的线程数。
Process	Private Bytes	专为此进程分配，无法与其他进程共享的当前字节数

Oracle资源监控

度量	描述
CPU used by this session	这是在用户调用开始和结束之间会话所占用的CPU时间(以10毫秒为单位)。一些用户调用在10毫秒之内即可完成，因此用户调用的开始和结束时间可以是相同的，在这种情况下，系统值为0毫秒，操作系统报告中可能有类似的问题，尤其是在经历许多上下文切换的系统中
Bytes received via SQL*Net from client	通过Net8从客户端接收的总字节数
Logons current	当前的登录总数
Opens of replaced files	由于已经不在进程文件缓存中，所以需要重新打开的文件总数
User calls	在每次登陆、解析或执行时。Oracle会分配资源（call　S tate对象）以记录相关的用户调用数据结构。在确定活动时，用户调用与RPI调用的比指明了因用户发往Oracle的请求类型而生成的内部工作量。
SQL*Net roundtrips to/from client	发送到客户端和从客户端接受的Net8消息的总数
Bytes sent via SQL*Net to client	从前台进程中发送到客户端的总字节数
Opened cursors current	当前打开的光标总数
DB block changes	由于与一致更改的关系非常密切，此统计计算对SGA中所有块执行的、作为更新或删除操作一部分的更改总数。这些更改将生成重做日志项。如果事物被提交，将是对数据库的永久性更改．此统计是一个全部数据库作业的粗略指示，并且指出（可能在每事物级上）弄脏缓冲区的速率
Total file opens	由实例执行的文件打开总数，每个进程需要许多文件（控制文件、日志文件、数据库文件）以便针对数据库进行工作

Web服务器监控

•**Apache**

Measurement	Description
# Busy Servers	The number of servers in the Busy state
# Idle Servers	The number of servers in the Idle state
Apache CPU Usage	The percentage of time the CPU is utilized by the Apache server
Hits/sec	The HTTP request rate
KBytes Sent/sec	The rate at which data bytes are sent from the Web server

•**IIS**

Measurement	Description
Get Requests/sec	The rate at which HTTP requests using the GET method are made. Get requests are generally used for basic file retrievals or image maps, though they can be used with forms.
Post Requests/sec	The rate at which HTTP requests using the POST method are made. Post requests are generally used for forms or gateway requests.
Maximum Connections	The maximum number of simultaneous connections established with the Web service
Current Connections	The current number of connections established with the Web service
Not Found Errors/sec	The rate of errors due to requests that could not be satisfied by the server because the requested document could not be found. These are generally reported to the client as an HTTP 404 error code.
Private Bytes	The current number of bytes that the process has allocated that cannot be shared with other processes.

中间件服务器监控 – WebSphere

•**Run Time Resources**

Measurement	Description
MemoryFree	The amount of free memory remaining in the Java Virtual Machine
MemoryTotal	The total memory allocated for the Java Virtual Machine
MemoryUse	The total memory in use within the Java Virtual Machine

•**TransactionData**

Measurement	Description
NumTransactions	The number of transactions processed
ActiveTransactions	The average number of active transactions
TransactionRT	The average duration of each transaction
BeanObjectCount	The average number of bean object pools involved in a transaction
RolledBack	The number of transactions rolled back
Commited	The number of transactions committed
LocalTransactions	The number of transactions that were local
TransactionMethodCount	The average number of methods invoked as part of each transaction
Timeouts	The number of transactions that timed out due to inactivity timeouts
TransactionSuspended	The average number of times that a transaction was suspended

负载压力测试实施

我们将负载压力测试实施步骤概括为如下：

- 测试计划
- 测试需求分析
- 测试案例制定
- 测试环境、工具、数据准备
- 测试脚本录制、编写与调试
- 场景制定
- 测试执行
- 获取测试结果
- 结果评估与测试报告

测试计划

- 制定一个全面的测试计划是负载测试成功的关键。定义明确的测试计划将确保制定的方案能完成负载测试目标。
- 负载压力测试计划过程的4个步骤:
  - 分析应用程序

    对硬件和软件组件、系统配置以及典型的使用模型有一个透彻的了解。
  - 定义测试目标

    开始测试之前，应精确地定义想要实现的目标。
  - 计划方案实施

    确定如何实现测试目标 ，定义性能度量的范围，确定需要的Vuser的数量和类型 。
  - 检查测试目标

    测试计划应该基于明确定义的测试目标。如：度量最终用户响应时间、定义最优的硬件配置、检查可靠性、查看硬件或软件升级、评估新产品、确定瓶颈、度量系统容量等。

测试需求分析

- 测试需求就是应用需求的衍生，而且测试用例也必须覆盖所有的测试需求，否则，这个测试过程就是不完整的。主要有以下的几个关键点：
  - 测试的对象是什么，例如“被测系统中有负载压力需求的功能点包括哪些？”；“测试中需要模拟哪些部门用户产生的负载压力？”等问题；
  - 系统配置如何，例如“预计有多少用户并发访问？”；“用户客户端的配置如何？”；“使用什么样的数据库”；“服务器怎样和客户端通信？”；“网络设备的吞吐能力如何，每个环节承受多少并发用户？”等问题；
  - 应用系统的使用模式是什么，例如“使用在什么时间达到高峰期？”；“用户使用该系统是采用B/S运行模式吗？”等问题。

80～20原理测试强度估算

- 80～20原理：每个工作日中80％的业务在20％的时间内完成。
- 举一个例子来看80～20原理如何应用与测试需求分析。
  - 去年全年处理业务约100万笔，其中15％的业务处理中每笔业务需对应用服务器提交7次请求；其中70％的业务处理中每笔业务需对应用服务器提交5次请求；其余15％的业务处理中每笔业务需对应用服务器提交3次请求。根据以往统计结果，每年的业务增量为15％，考虑到今后3年业务发展的需要，测试需按现有业务量的两倍进行。每年业务量集中在8个月，每个月20个工作日，每个工作日8小时 。
  - 测试强度估算如下：
  - 每年总的请求数为：（100×15%×7+100×70%×5+100×15%×3）×2=1000万次/年
  - 每天请求数为：1000/160=6.25万次/天
  - 每秒请求数为：（62500×80%）/（8×20%×3600）=8.68次/秒
  - 即服务器处理请求的能力应达到9次/秒。

测试案例制定

<RDMonitor1.2性能测试案例>

本次并发性能测试主要模拟以下过程:

1. 模拟50用户使用浏览器同时并发访问rdmis系统登录页面login.jsp，并使用不同的用户名（user1～user50）进行登录。而每个虚拟用户对应不同的IP地址列表。
2. 登录后，可以选择以下两种策略:
   - 在脚本中设置集合点。虚拟用户在登录进rdmis系统之后，先暂停，等待所有的虚拟用户都登录进来之后，再并发进行后续的操作。设置集合点的目的在于，使所有的虚拟用户在登录到rdmis系统之后，能够同时并发访问流量信息统计查询页面。
   - 在脚本中不设置集合点。先登录到rdmis里来的用户会继续进行后续的操作，而不等待其它用户。这种情况下，在同一时刻压力分散到各个模块中，给流量查询模块带来的压力有所下降。

测试案例制定

3. 在执行上述两种策略之一后，进入流量信息统计查询页面。每个虚拟用户查询自己IP地址列表所对应的流量信息。
4. 在上述操作完成后，虚拟用户点击页面中的“详细信息”链接进入下一层查询页面。在这里，由于每个虚拟用户的“详细信息”链接的内容是不同的。
5. 在上述操作完成后，虚拟用户点击页面中的“详细信息”链接进入第三层查询页面。同样需要改写测试脚本中“详细信息”链接里IP地址的值为动态获得的值。
6. 退出系统，整个操作完成。
7. 在本次测试中，并发用户数设置为50，每个虚拟用户只执行一次脚本而不循环多次。

测试环境、工具、数据准备

- 测试环境直接影响测试效果，所有的测试结果都是在一定软硬件环境约束下的结果，测试环境不同，测试结果可能会有所不同，需要注意：
  - 如果是完全真实的应用运行环境，要尽可能降低测试对现有业务的影响；
  - 如果是建立近似的真实环境，要首先达到服务器、数据库以及中间件的真实，并且要具备一定的数据量，客户端可以次要考虑；
- 古人云“工欲善其事，必先利其器”，一个测试工具能否满足测试需求、能否达到令人满意的测试结果是选择测试工具要考虑的最基本的问题。QALoad、LoadRunner 都是不错的选择。
- 实施负载压力测试时，需要运行系统相关业务，这时需要一些数据支持才可运行业务，这部分数据即为初始测试数据。有时为了模拟不同的虚拟用户的真实负载，需要将一部分业务数据参数化，这部分数据为参数化测试数据。

场景制定

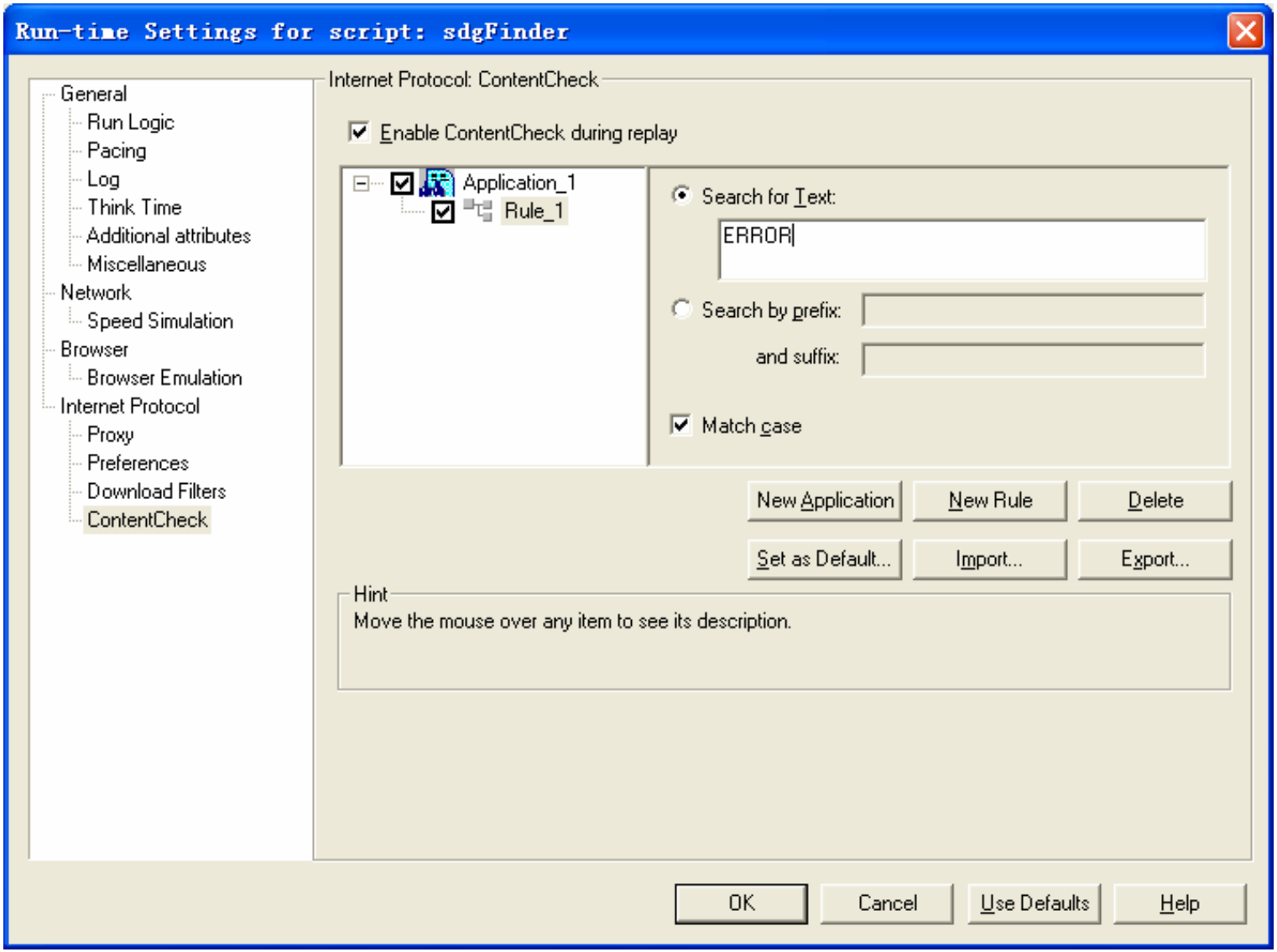
- 创建Vuser组

  Vuser 组用于将方案中的 Vuser 组织成可管理的组。
- 配置Vuser组中的Vuser

  对于每个Vuser，可以分配不同的脚本和负载生成器计算机。
- 配置Vuser运行时的设置

  可以设置脚本的运行时设置，采用在控制中心自定义执行 Vuser 脚本的方式。如：是否加入思考时间、超时时间的设置等等。
- 配置负载生成器

  在测试执行之前，需要配置方案的负载生成器和 Vuser 行为，即制定场景。如：执行脚本的先后顺序，执行脚本的循环次数等等。


Run-time Settings for script: sdgFinder
General
Run Logic
Pacing
Log
Think Time
Additional attributes
Miscellaneous
Network
Speed Simulation
Browser
Browser Emulation
Internet Protocol
Proxy
Preferences
Download Filters
ContentCheck
Internet Protocol: ContentCheck
Enable ContentCheck during replay
Application_1
Rule_1
Search for Text:
ERROR
Search by prefix:
and suffix:
Match case
New Application
New Rule
Delete
Set as Default...
Import...
Export...
Hint
Move the mouse over any item to see its description.
OK
Cancel
Use Defaults
Help

测试执行

- 运行场景

  运行场景时，会为 Vuser 组分配负载生成器并执行它们的 Vuser 脚本。在场景运行时，可以监视每个 Vuser、查看由 Vuser 生成的错误、警告和通知消息以及停止 Vuser 组和各个 Vuser。可以允许单个 Vuser 或组中的 Vuser 在停止前完成它们正在运行的迭代、在停止前完成它们正在运行的操作或者立即停止运行。

- 在执行期间查看Vuser

  可以在场景执行期间查看 Vuser 的活动：在 Controller 负载生成器计算机中，可以查看输出窗口、联机监视 Vuser 性能，以及查看执行场景的 Vuser 的状态；

- 监视场景

  “运行时”监视器显示参与场景的 Vuser 的数目和状态，以及 Vuser 所生成的错误数量和类型。 “事务”监视器显示场景执行期间的事务速率和响应时间。“资源”监视器用于度量场景运行期间各种服务器资源的统计信息。

结果评估与测试报告

- 交易处理性能评估
  - 并发用户数

    并发用户数是负载压力测试的主要指标，体现了系统能够承受的并发性能。

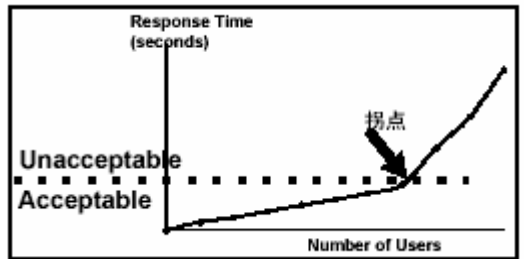
    测试重点得到两类并发用户数指标，一类是系统最佳性能的并发用户数，另一类是系统能够承受的最大并发用户数，这两类指标在某种情况下有可能重叠。
  - 交易响应时间

    该指标描述交易执行的快慢程度，这是用户最直接感受到的系统性能，也是故障定位迫切需要解决的问题。
  - 交易通过率

    指每秒钟能够成功执行的交易数，描述系统能够提供的“产量”，用户可以以此来评估系统的性能价格比。
  - 吞吐量

    指每秒通过的字节数，以及通过的总字节数。此指标在很大程度上影响系统交易的响应时间，形成响应时间的“拐点”。
  - 点击率

    描述系统响应请求的快慢。



结果评估与测试报告

- 系统资源占用性能评估
  - Available Mbytes：可用物理内存数。
  - page/sec：频繁的页交换将降低系统性能。一般如果pages/sec持续高于几百，那么应该进一步研究页交换活动。
  - Disk Time：指所选磁盘驱动器忙于为读或写入请求提供服务所用的时间的百分比。如果只有%Disk Time比较大，而CPU和内存都比较适中，硬盘可能会是瓶颈。
  - Avg.Disk Queue Length：该值应不超过磁盘数的1.5～2 倍。
  - 由于过多的页交换要使用大量的硬盘空间，因此有可能将导致将页交换内存不足与导致页交换的磁盘瓶颈混淆。
  - 如果页面读取操作速率很低，同时 Disk Time 和 Avg.Disk Queue Length的值很高，则可能有磁盘瓶颈；而如果队列长度增加的同时页面读取速率并未降低，则内存不足。
  - Processor Time： 如果该值持续超过95%，表明瓶颈是CPU。
  - Privileged Time：（CPU内核时间）是在特权模式下处理线程执行代码所花时间的百分比。如果该参数值和“Physical Disk”参数值一直很高，表明I/O有问题。

负载压力测试技巧

- 参数池

  使用指定的数据源中的值来替换原值。该数据源可以是一个文件或者内部生成的变量。
- 将事务插入到Vuser脚本

  可以定义事务以度量服务器的性能。要度量事务，需要插入任务的开始和结束标记。
- 将集合点插入到Vuser脚本

  要在系统上模拟较重的用户负载，需要同步各个 Vuser 以便在同一时刻执行任务。通过创建集合点，可以确保多个 Vuser 同时执行操作。
- 动态关联

  利用测试工具的脚本函数可以关联动态且不可人工预知的值。一些系统的输出值需要为后续操作提供输入，这些值只对当前会话有效，如SessionID等。
- IP模拟

  可以在负载生成器计算机上定义多个IP 地址，以模拟用户使用不同计算机的真实情况。

讨论和演示

- 测试需求：考察sdgfinder系统在测试环境上，50用户并发时，“查询”操作的平均响应时间是否小于10秒，操作系统资源占用情况是否存在瓶颈。
- 请根据前面讲到的负载压力测试实施步骤，完成本次性能测试从测试计划、测试实施到收集测试结果的全过程。在测试中要尽最大可能模拟真实业务发生情况，并运用参数池、事务、集合点、动态关联等技术。
- 测试中使用LoadRunner作为测试工具。

谢谢大家！