

# 软件性能测试工具综述

中国工商银行股份有限公司数据中心(北京) 朱怡雯 钱超 林勇

众所周知,性能测试工具一般通过Winsock和HTTP等协议记录用户操作。而协议的选择是基于软件的系统架构,不同的性能测试工具,脚本语言也不同。所以开展性能测试需要对各种性能测试工具进行评估,每一种性能测试工具都有自身的特点,只有经过工具评估,才能选择符合现有软件架构的性能测试工具。因此确定使用什么样的测试工具是性能测试的关键一步。由于性能测试工具涉及的内容很多,本文只能对性能测试工具作一般性介绍,重点讨论性能测试工具的基本功能和可能出现的技术要点。

## 一、性能测试工具概述

### 1.服务器整机系统性能测试工具

一台服务器的系统性能可以按照处理器、内存、存储、网络等部分来划分,而针对不同的应用,可能会对某些部分的性能要求更高一些。

#### (1)Iometer:存储子系统读写性能测试

Iometer是Windows系统下对存储子系统的读写性能进行测试的软件。可以显示磁盘系统的最大I/O能力、磁盘系统的最大吞吐量、CPU使用率、错误信息等。用户可以通过设置不同的测试参数,如存取类型(Sequential、Random等)、读写块大小(64K、256K等)、队列深度等,来模拟实际应用的读写环境来进行测试。

#### (2)Iozone:Linux下I/O性能测试

现在有很多的服务器系统采用Linux操作系统,在

Linux平台下测试I/O性能可以采用Iozone。Iozone是一个文件系统的Benchmark工具,可以测试不同的操作系统中文件系统的读写性能。可以测试read, write, re-read, re-write, readbackwards, readstrided, fread, fwrite, randomread, pread, mmap, aio\_read, aio\_write等不同模式下的硬盘性能。

#### (3)Netperf:网络性能测试

Netperf可以测试服务器网络性能,针对基于TCP或UDP的传输。Netperf根据应用的不同,可以进行不同模式的网络性能测试,即批量数据传输(Bulk Data Transfer)模式和请求/应答(Request/Response)模式。Netperf测试结果所反映的是,一个系统能够以多快的速度向另外一个系统发送数据,以及另外一个系统能够以多快的速度接收数据。

## 2.测试工具的应用

这里所谈的是针对应用系统的测试工具。应用系统的性能是一个复杂的概念,覆盖面非常广泛,对一个软件系统而言包括执行效率、资源占用率、稳定性、安全性、兼容性、可靠性等,目前市场上的性能测试工具种类很多,一般可以划分为以下几种:负载压力测试工具、资源监控工具、故障定位工具以及调优工具。下面重点从负载压力方面来介绍,主流的负载性能测试工具有以下几个。

(1)QALoad:美国Compuware公司的QALoad是客户机/服务器系统、企业资源计划(ERP)和电子商务应用

的自动化负载测试工具。QALoad是QACenter性能版的一部分,它通过可重复的、真实的测试能够彻底地度量应用的可扩展性和性能。

(2)LoadRunner: LoadRunner是一种预测系统行为和性能的负载测试工具,通过模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题。LoadRunner能够对整个企业架构进行测试,LoadRunner适用于各种体系架构,能支持广范的协议和技术(如Web、FTP、Database等),能预测系统行为并优化系统性能。

(3)Webload: Webload是RadView公司推出的一个性能测试和分析工具,它让Web应用程序开发者自动执行压力测试;Webload通过模拟真实用户的操作,生成压力负载来测试Web应用的性能。

这些软件的功能虽然强大,并且可以自动生成测试报告,但其终究是一个工具,如果想真正定位服务器性能的好坏和性能的瓶颈所在,需要使用工具的人对于测试软件的方方面面都要有了解,比如软件体系构架、网络拓扑、服务器硬件等知识。

## 二、性能工具的应用与实例

典型的业界系统的架构大多由客户机、网络、防火墙、负载均衡器、Web服务器、应用服务器(中间件)、数据库等环节组成。根据木桶原理,即木桶所能装的水的量取决于最短的那块木板,整个系统的性能要得到提高,每个环节的性能都需要优化。对于一个庞大的系统来说,除了测试期间测试工作充分的准备以外,如何用好对测试工具就更为重要。

笔者以LoadRunner性能测试工具为例进一步加以说明,系统架构如图1所示。

### 1.性能测试工具的组成

性能测试工具有四个组成部分:虚拟用户脚本产生器(Vuser Generator),压力调度、监控系统(Controller),压力产生器(Load Generator),数据分析工具(Analysis)。

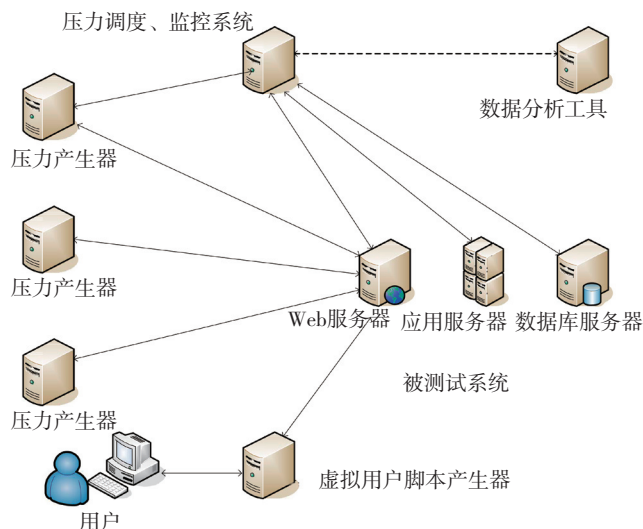


图1 性能测试系统架构

虚拟用户脚本产生器用来录制脚本,实际上是一套开发调试工具。压力调度、监控系统是一个框架程序和监控程序,它负责将虚拟用户脚本产生器开发的脚本以多进程/多线程的方式在压力产生器服务器上运行。为了产生更大的压力,压力调度、监控系统必需支持集群功能,理论上压力调度、监控系统可以和任意多台压力产生器服务器互联,以便产生足够大的负载压力。压力调度、监控系统同时实现无代理方式的监控功能,可以监控各种主流的软件,并且提供对不支持的软件进行监控的二次开发的手段。数据分析工具用于事后的数据分析,可以安装在任何Windows平台的服务器上。

#### (1)虚拟用户脚本产生器

虚拟用户脚本产生器通过录制客户机和后台服务器之间的通信包,分析其中的协议,自动产生脚本。用户在自动产生的脚本的基础上进行修改,从而快速开发出一个逻辑功能和客户机软件完全一样的压力脚本程序。

虚拟用户脚本产生器的体系结构分为三部分:第一部分为底层Proxy录制器,负责捕获客户机和服务器之间通信的数据包。第二部分是界面部分,提供脚本编辑、调试和运行功能,这部分可以用Visual C++/MFC实现

Windows平台版本和Java/AWT实现Unix版本。第三部分是以插件的形式提供的分析各种网络协议的解析器。

录制的技术主要是通过Proxy的方式来实现的。

虚拟用户脚本产生器根据对捕获的数据的分析,将其还原成对应协议的API组成的脚本。由于Proxy源程序的获得非常容易,虚拟用户脚本产生器的主要技术要点是,如何根据捕获的数据包来反解析成对应的网络协议。通常捕获的数据包为TCP数据流,我们可以很容易的生成socket层次的脚本,类似如下示例:

```
Action()
{
    lr_think_time(13);
    lrs_send("socket0","buf12",LrsLastArg);
    lrs_receive("socket0","buf13",LrsLastArg);
    lrs_send("socket0","buf14",LrsLastArg);
    lrs_receive("socket0","buf15",LrsLastArg);
    lrs_send("socket0","buf16",LrsLastArg);
    lrs_receive("socket0","buf17",LrsLastArg);
    return0;
}
```

其中date.ws包含着录制时捕获的数据包,按照“发-收-发-收-发-收”的顺序排放。这样的脚本按照记录的收发过程来回放,但它的最大的缺点是处于太底层。

当然,socket方式是一切应用层协议的基础,socket脚本是一种通用的方式。对于Vuser Generator不支持的应用层协议只能通过socket层次来录制。因此Vuser Generator能生成socket-API脚本是其最基本的功能。

客户机和服务器之间的通信,有一部分是数据是动态的,每次通信都不一样。录制器在录制的时候是无法区分哪些是静态的信息,哪些是动态的信息,所有的信息都以hard-coded的方式记录下来。但是在回放的时候,如果有些信息不做关联,脚本是不能成功执行的,例如下面代码中的阴影标注字体部分。解决方法是:第一次从服务器得到SessionID后,将其放在变量{PeopleSoftsessionID2}里面,在后面脚本访问服务

器的语句里面,把所有的需要关联的地方替换为变量{PeopleSoftsessionID2}调用其值就可以顺利地往下执行脚本了。

```
Action()
{
    web_url("pcm_df_badloan_licence_apply_main.jsp",
        "URL=http://83.33.14.81/pcm/d/df/pcm_df_badloan_licence_apply_main.jsp?kindflag=0&dse_sessionId={PeopleSoftsessionID2}",
        ..... "Referer=http://**.**.14.81/includes/menu.jsp?dse_sessionId={PeopleSoftsessionID2}",
        .....
        lr_start_transaction("***列表查询");
        ..... "Referer=http://**.**.14.81/pcm/d/df/pcm_df_badloan_licence_apply_main.jsp?kindflag=0&dse_sessionId={PeopleSoftsessionID2}",
        "Snapshot=t10.inf",
        .....
        lr_end_transaction("***列表查询",LR_AUTO);
        return0;
}
```

服务器返回给客户机一些动态变化的值,客户机使用这些值去访问服务器的时候,不能把这些值写死在脚本里面,而应该存放在一个变量里面。这就是关联的概念。我们针对每一个具体的系统进行细致的分析,确定其需要关联的动态信息。在平时的工作中常用到的自动关联的方法有三种:①在录制之前设定辨别规则,录制完毕,产生脚本的时候根据规则识别出需要关联的动态内容,从而产生正确的脚本。②录制完毕回放一遍,把回放结果与录制结果进行自动对比,确定动态信息,进行自动关联。③录制两个一模一样的脚本,对比其中的差异来确定需要关联的动态信息,然后进行关联。

## (2)压力调度、监控系统和压力产生器

压力调度、监控系统是整个压力测试的核心。压力产生器是产生压力的负载产生器,它们以进程或者线程

的方式运行由虚拟用户脚本产生器生成脚本。压力调度、监控系统和压力产生器实际上是一套框架程序。具体执行什么功能,是由脚本来完成的。

压力调度、监控系统包括若干进程/线程。每种进程的作用如下:Center进程是整个调度的核心进程,它负责联系和用户界面打交道的工作;Agent进程负责和远端的压力产生器服务器中对应的Agent进程通信。负责把编译好的脚本传送到压力产生器服务器上。在脚本运行的时候,定期从压力产生器服务器上获取压力产生器的运行状态,每个虚拟用户运行的日志;Monitor进程负责对被测系统的各个环节进行监控,并把监控的内容写入压力调度、监控系统服务器的本地磁盘,同时把监控的内容传送给Center进程,实时地显示在用户界面上(如图2所示)。

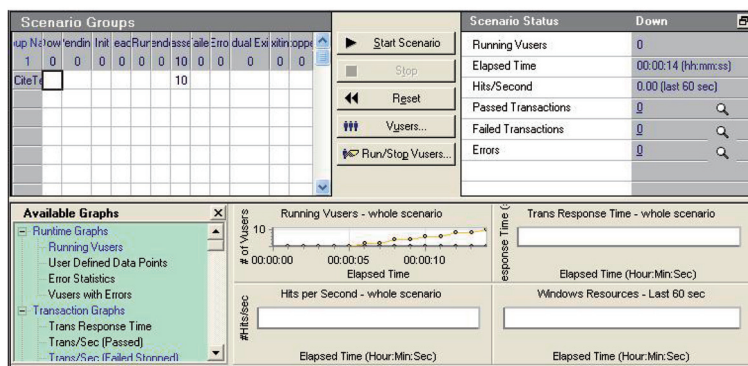


图2 用户界面

压力产生器的进程有两种,一个是Agent进程,另一个是压力产生器进程。Agent进程负责和压力调度、监控系统服务器通信,它根据压力调度、监控系统的指示,在本服务器上派生出指定数目的压力产生器进程,这些压力产生器进程负责具体执行相应的脚本。压力产生器进程个数就是虚拟用户的个数。

在操作中最大的难点是监控问题。当把被测系统的各个环节都监控起来,要监控的参数会有成百上千个。如果采用集中式监控的方式,采集数据本身会对系统造成很大的影响,所以必需支持分布式监控方式。由于采

集的数据是来自不同服务器上的,由于各种的延迟,数据之间的时间同步将是一个重大的问题。

### (3)数据分析工具

该工具是一个纯数学工具软件,可以将压力产生的数据直接导入其中进行处理。所以只要提供开放的数据接口就可以了,无需自己开发独立的性能数据分析软件。

## 2.性能测试的一般步骤

通常,进行性能测试项目的一般步骤如下:

(1)用户确定需要录制的交易,通过用户操作和虚拟用户脚本产生器的录制,记录并生成自动化脚本。

(2)修改脚本,确定脚本能够回放成功。

(3)压力调度、监控系统是一个集中控制平台,它和压力产生器互联,指定脚本在压力产生器上的分配,并控制压力产生器向被测系统的加压方式和行为。

(4)压力调度、监控系统同时负责搜集被测系统的各个环节的性能数据。各个压力产生器会记录最终用户响应时间和脚本执行的日志。

(5)压力运行结束以后,压力产生器将数据传送到压力调度、监控系统中,压力调度、监控系统负责将数据汇总。

(6)数据分析工具读取压力测试数据,进行分析工作,确定瓶颈和调优方法。

(7)针对性地进行系统调优,重复进行压力测试,确定性能是否得到提高。

重复以上3~7步,可以逐步提高系统的性能。 **FCC**