

具体实例教你如何做 LoadRunner 结果分析

1. 前言:

LoadRunner 最重要也是最难理解的地方--测试结果的分析.其余的录制和加压测试等设置对于我们来讲通过几次操作就可以轻松掌握了.

针对 Results Analysis 我用图片加文字做了一个例子,希望通过例子能给大家更多的帮助.

这个例子主要讲述的是多个用户同时接管任务,测试系统的响应能力,确定系统瓶颈所在.客户要求响应时间是 1 个人接管的时间在 5S 内.

2. 系统资源:

2.1 硬件环境:

CPU: 奔四 2.8E

硬盘: 100G

网络环境: 100Mbps

2.2 软件环境:

操作系统: 英文 windowsXP

服务器: tomcat 服务

浏览器: IE6.0

系统结构: B/S 结构

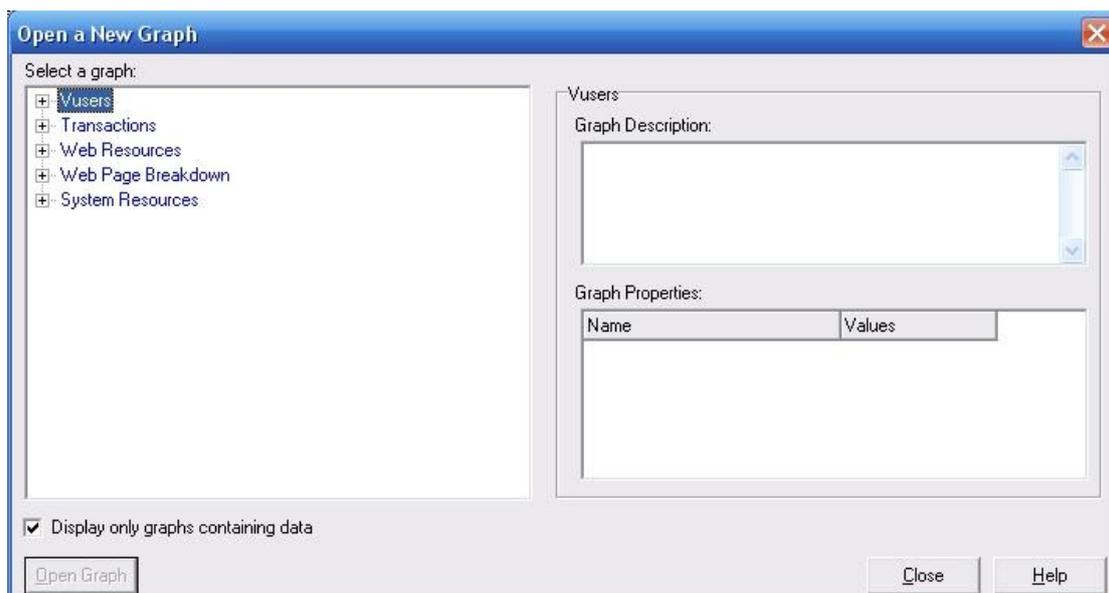
3. 添加监视资源

下面要讲述的例子添加了我们平常测试中最常用到的一些资源参数.另外有些特殊的资源暂时在这里不做讲解了.我会在以后相继补充进来。

Mercury Loadrunner Analysis 中最常用的 5 种资源.

1. Vuser
2. Transactions
3. Web Resources
4. Web Page Breakdown
5. System Resources

在 Analysis 中选择 “Add graph” 或 “New graph” 就可以看到这几个资源了. 还有其他没有数据的资源,我们没有让它显示.



如果想查看更多的资源,可以将左下角的 display only graphs containing data 置为不选.然后选中相应的点“open graph”即可.

打开 Analysis 首先可以看的是 Summary Report.这里显示了测试的分析摘要.应有尽有.但是我们并不需要每个都要仔细去看.下面介绍一下部分的含义:

Duration(持续时间): 了解该测试过程持续时间.测试人员本身要对这个时期内系统一共做了多少的事有大致地熟悉了解.以确定下次增加更多的任务条件下测试的持续时间。

Statistics Summary(统计摘要): 只是大概了解一下测试数据,对我们具体分析没有太大的作用.

Transaction Summary(事务摘要): 了解平均响应时间 Average 单位为秒.其余的看不看都可以.都不是很重要.

4. 分析集合点

在录制脚本中通常会使用到集合点,那么既然我们用到了集合点,我们就需要知道 Vuser 是在什么时候集合在这个点上,又是怎样的一个被释放的过程.这个时候就需要观察 Vuser-Rendezvous 图.

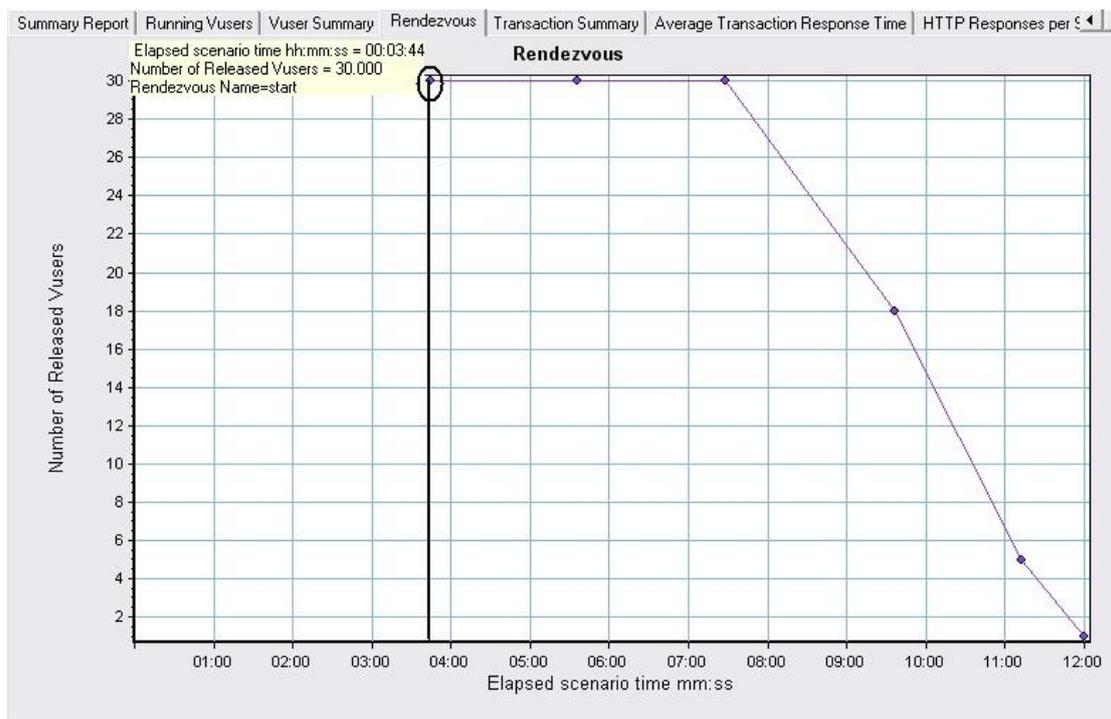


图 1

可以看到大概在 3 分 50 的地方 30 个用户才全部集中到 start 集合点,持续了 3 分多,在 7 分 30 的位置开始释放用户,9 分 30 还有 18 个用户,11 分 10 还有 5 个用户,整个过程持续了 12 分.

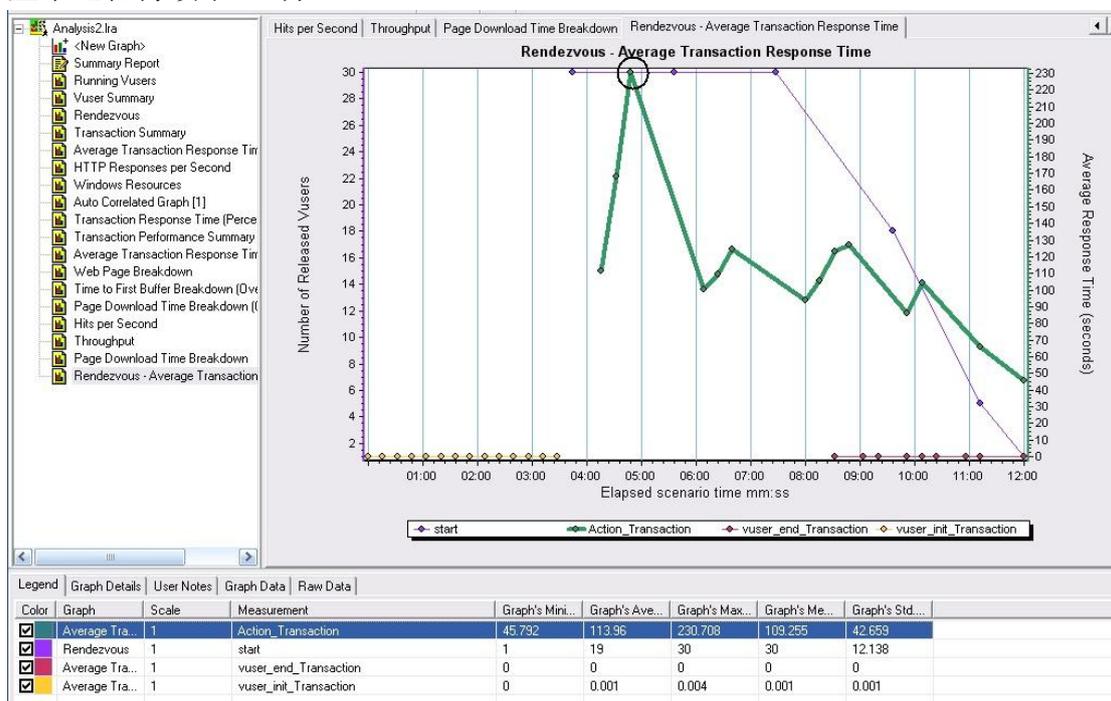


图 2

上面图 2 是集合点与平均事务响应时间的比较图.

注:在打开 analysis 之后系统 LR 默认这两个曲线是不在同一张图中的.这就需要自行设置了.具体步骤如下:

点击图上.右键选择 merge graphs.然后在 select graph to merge with 中选择即

将用来进行比较的 graph.如图 3:

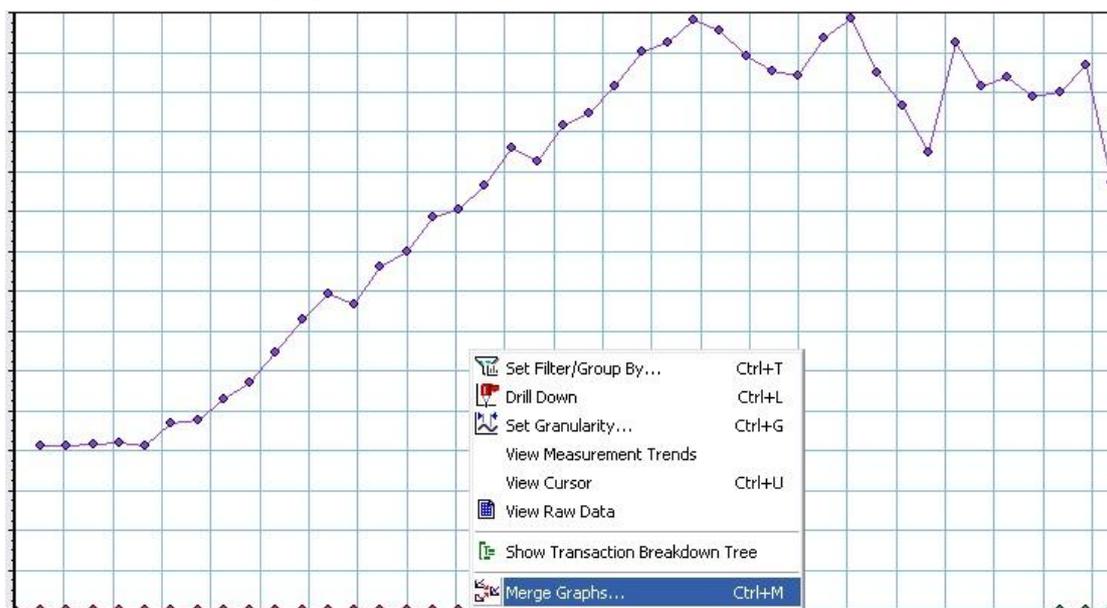


图 3

图 2 中较深颜色的是平均响应时间,浅色的为集合点,当 Vuser 在集合点持续了 1 分后平均响应时间呈现最大值,可见用户的并发对系统的性能是一个很大的考验.接下来看一下与事务有关的参数分析.下看一张图.

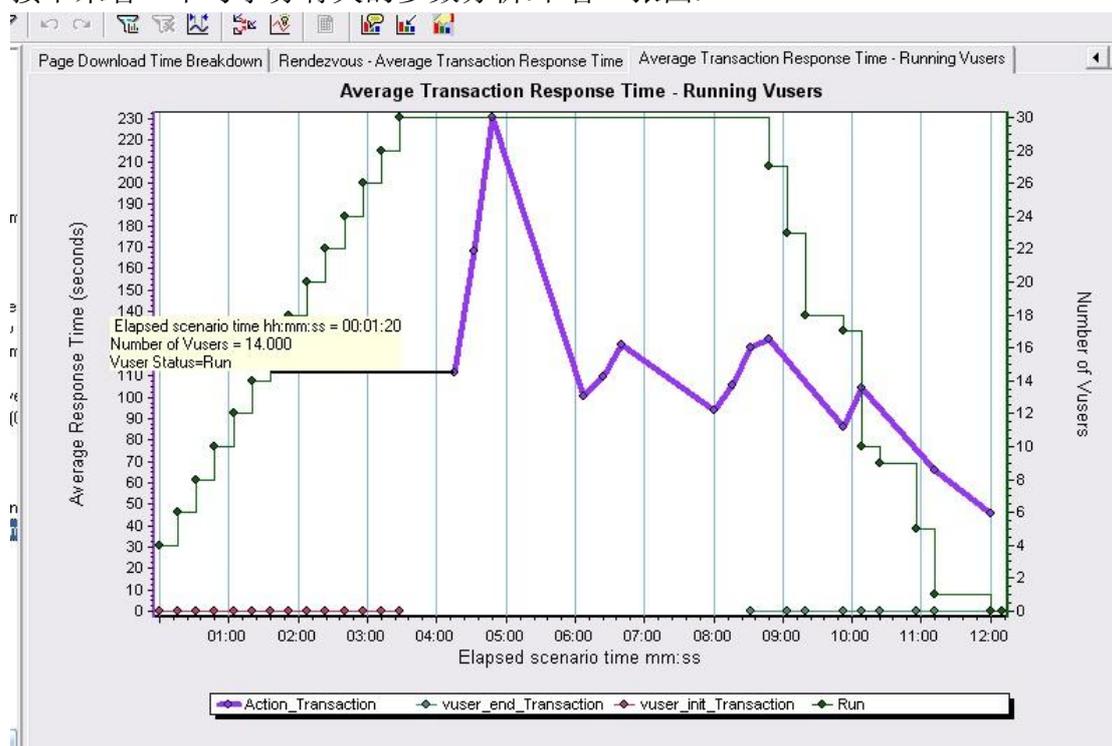
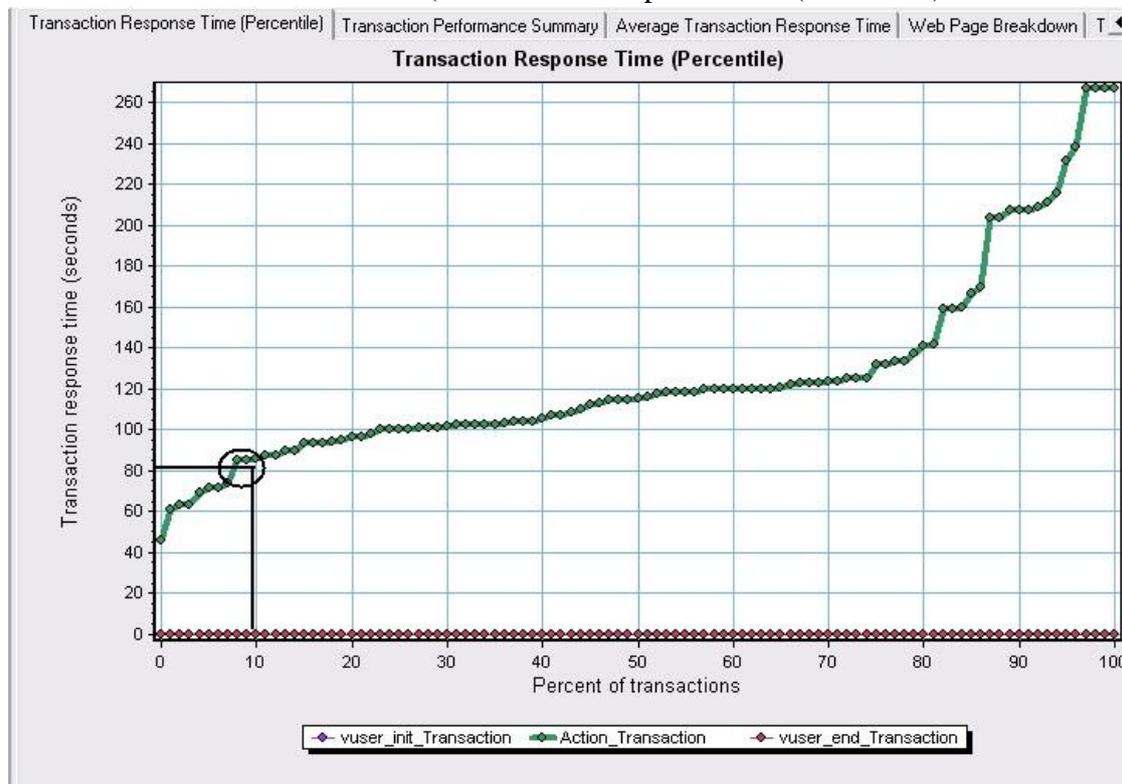


图 4

这张图包括 Average Transaction Response Time 和 Running Vuser 两个数据图.从图中可以看到 Vuser_init_Transaction(系统登录)对系统无任何的影响,Vuser 达到 15 个的时候平均事务响应时间才有明显的升高,也就是说系统达到最优性能的时候允许 14 个用户同时处理事务,Vuser 达到 30 后 1 分,系统响应时间最大,那么这个最大响应时间是要推迟 1 分钟才出现的,在系统稳定之后事务响应时间开始

下降说明这个时候有些用户已经执行完了操作.同时也可以看出要想将事务响应时间控制在 10S 内.Vuser 数量最多不能超过 2 个.看来是很难满足用户的需求了.

做一件事有时候上级会问你这件事办得怎么样了.你会说做完一半了.那么这个一半的事情你花了多少时间呢?所以我们要想知道在给定时间的范围内完成事务的百分比就要靠下面这个图(Transaction Response Time(Percentile))

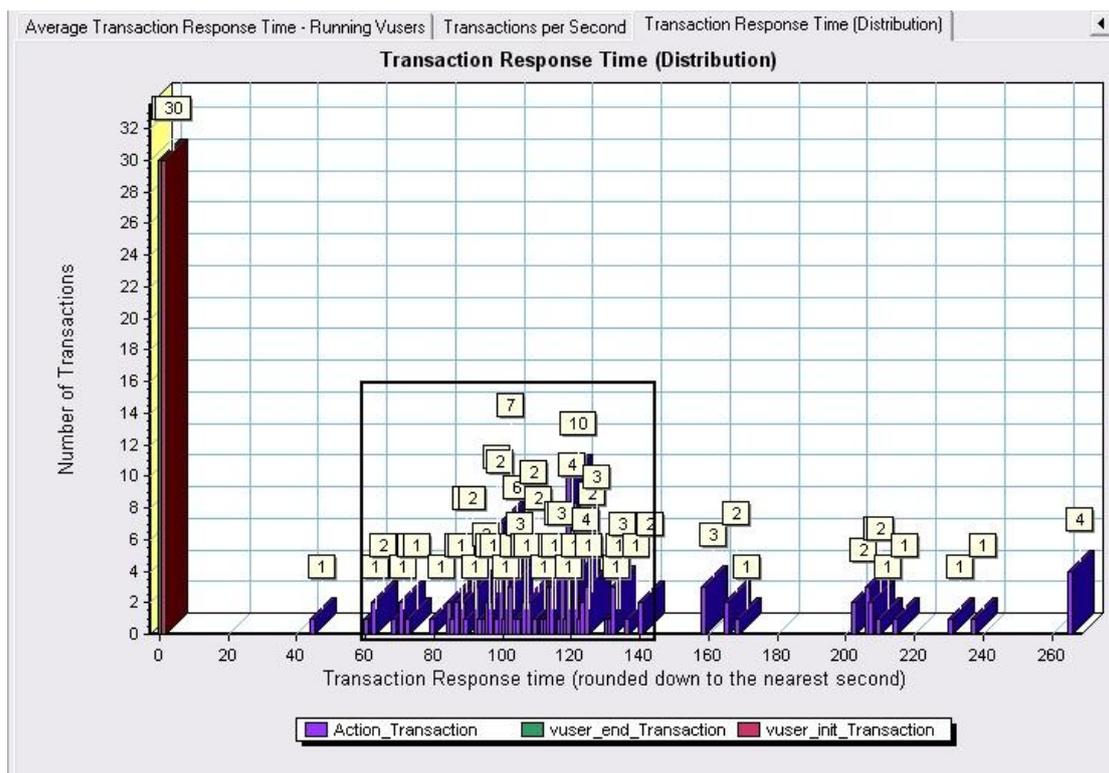


图中画圈的地方表示 10% 的事务的响应时间是在 80S 左右.80S 对于用户来说不是一个很小的数字,而且只有 10% 的事务,汗.你觉得这个系统性能会好么!

实际工作中遇到的事情不是每一件事都能够在很短的时间内完成的,对于那些需要时间的事情我们就要分配适当的时间处理,时间分配的不均匀就会出现有些事情消耗的时间长一些,有些事情消耗的短一些,但我们自己清楚.LR 同样也为我们提供了这样的功能,使我们可以了解大部分的事务响应时间是多少?以确定这个系统我们还要付出多少的代价来提高它.

Transaction Response Time(Distribution)-事务响应时间(分布)

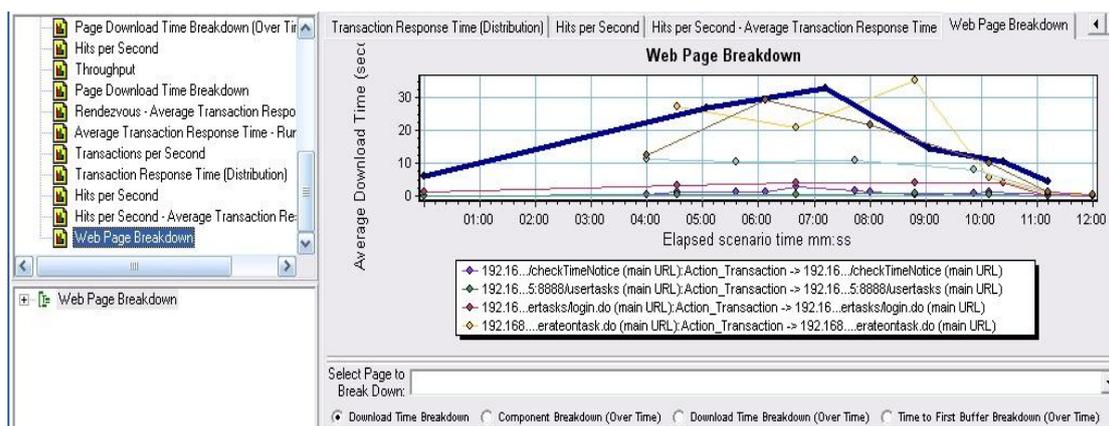
显示在方案中执行事务所用时间的分布.如果定义了可以接受的最小和最大事务性能时间,可以通过此图确定服务器性能是否在可接受范围内.



很明显大多数事务的响应时间在 60-140S.在我测试过的项目中多数客户所能接受的最大响应时间也要在 20S 左右.140S 的时间!很少有人会去花这么多的时间去等待页面的出现吧!

通过观察以上的数据表.我们不难看出此系统在这种环境下并不理想.世间事有果就有因,那么是什么原因导致得系统性能这样差呢?让我们一步一步的分析.

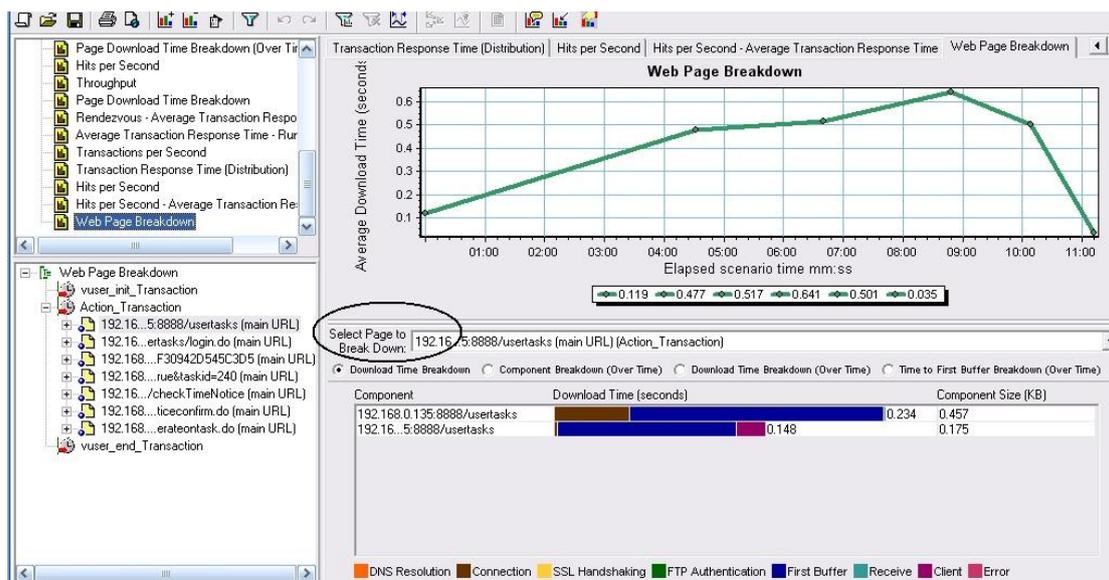
系统性能不好的原因多方面,我们先从应用程序看.有的时候我不得不承认 LR 的功能真的很强大,这也是我喜欢它的原因.先看一张页面细分图.



一个应用程序是由很多个组件组成的,整个系统性能不好那我们就把它彻底的剖析一下.图片中显示了整个测试过程中涉及到的所有 web 页.web page breakdown 中显示的是每个页面的下载时间.点选左下角 web page breakdown 展开,可以看到每个页中包括的 css 样式表,js 脚本,jsp 页面等所有的属性.

在 select page to breakdown 中选择页面.

见图.



在 Select Page To Breakdown 中选择 <http://192.168.0.135:8888/usertasks> 后,在下方看到属于它的两个组件,第一行中 Connection 和 First Buffer 占据了整个的时间,那么它的消耗时间点就在这里,我们解决问题就要从这里下手。

名称	描述
DNS Resolution	显示使用最近的 DNS 服务器将 DNS 名称解析为 IP 地址所需的时间。“DNS 查找”度量是指示 DNS 解析问题或 DNS 服务器问题的一个很好的指示器。
Connection	显示与包含指定 URL 的 Web 服务器建立初始连接所需的时间。连接度量是一个很好的网络问题指示器。此外,它还可表明服务器是否对请求作出响应。
First Buffer	显示从初始 HTTP 请求(通常为 GET)到成功收回来自 Web 服务器的第一次缓冲时为止所经过的时间。第一次缓冲度量是很好的 Web 服务器延迟和网络滞后指示器。 注意:由于缓冲区大小最大为 8K,因此第一次缓冲时间可能也就是完成元素下载所需的时间。
SSL Handshaking	显示建立 SSL 连接(包括客户端 hello、服务器 hello、客户端公用密钥传输、服务器证书传输和其他部分可选阶段)所用的时间。自此点之后,客户端与服务器之间的所有通信都将被加密。 SSL 握手度量仅适用于 HTTPS 通信。
Receive	显示从服务器收到最后一个字节并完成下载之前经过的时间。 “接收”度量是很好的网络质量指示器(查看用来计算接收速率的时间/大小比率)。
FTP Authentication	显示验证客户端所用的时间。如果使用 FTP,则服务器在开始处理客户端命令之前,必须验证该客户端。 “FTP 验证”度量仅适用于 FTP 协议通信。

 Client	显示因浏览器思考时间或其他与客户端有关的延迟而使客户机上的请求发生延迟时，所经过的平均时间。
 Error	显示从发出 HTTP 请求到返回错误消息（仅限于 HTTP 错误）这期间经过的平均时间。

也有可能你的程序中 client 的时间最长.或者其他的,这些就要根据你自己的测试结果来分析了.下面我们来看一下 CPU,内存.硬盘的瓶颈分析方法:

首先我们要监视 CPU,内存.硬盘的资源情况.得到以下的参数提供分析的依据.

%processor time(processor_total):器消耗的处理时间数量.如果服务器专用于 sql server 可接受的最大上限是 80% -85 %.也就是常见的 CPU 使用率.

%User time(processor_total)::表示耗费CPU的数据库操作，如排序，执行aggregate functions 等。如果该值很高，可考虑增加索引，尽量使用简单的表联接，水平分割大表格等方法来降低该值。

%DPC time(processor_total):: 越低越好。在多处理器系统中，如果这个值大于50%并且 Processor:% Processor Time非常高，加入一个网卡可能会提高性能，提供的网络已经不饱和。

%Disk time(physicaldisk_total):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。如果三个计数器都比较大，那么硬盘不是瓶颈。如果只有%Disk Time比较大，另外两个都比较适中，硬盘可能会是瓶颈。在记录该计数器之前，请在Windows 2000 的命令窗口中运行diskperf -yD。若数值持续超过80%，则可能是内存泄漏。

Available bytes(memory):用物理内存数。如果Available Mbytes的值很小（4 MB 或更小），则说明计算机上总的内存可能不足，或某程序没有释放内存。

Context switch/sec(system): (实例化inetinfo 和dllhost 进程) 如果你决定要增加线程字节池的大小，你应该监视这三个计数器（包括上面的一个）。增加线程数可能会增加上下文切换次数，这样性能不会上升反而会下降。如果十个实例的上下文切换值非常高，就应该减小线程字节池的大小。

%Disk reads/sec(physicaldisk_total):每秒读硬盘字节数.

%Disk write/sec(physicaldisk_total):每秒写硬盘字节数.

Page faults/sec:进程产生的页故障与系统产生的相比较，以判断这个进程对系统页故障产生的影响。

Pages per second:每秒钟检索的页数。该数字应少于每秒一页

Working set:理线程最近使用的内存页，反映了每一个进程使用的内存页的数量。如果服务器有足够的空闲内存，页就会被留在工作集中，当自由内存少于一个特定的阈值时，页就会被清除出工作集。

Avg.disk queue length:读取和写入请求(为所选磁盘在实例间隔中列队的)的平均数。该值应不超过磁盘数的1.5~2 倍。要提高性能，可增加磁盘。注意：一个Raid Disk实际有多个磁盘。

Average disk read/write queue length: 指读取(写入)请求(列队)的平均数

Disk reads/(writes)/s:理磁盘上每秒钟磁盘读、写的次数。两者相加，应小于磁盘设备最大容量。

Average disk sec/read:以秒计算的在此盘上读取数据的所需平均时间。

Average disk sec/transfer:指以秒计算的在此盘上写入数据的所需平均时间。

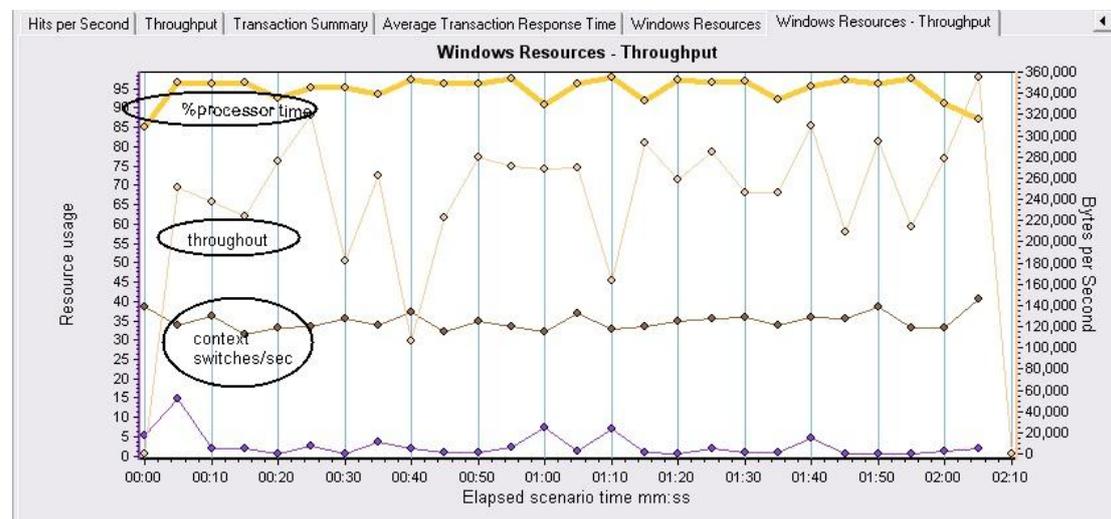
Bytes total/sec:为发送和接收字节的速率，包括帧字符在内。判断网络连接速度是否是瓶颈，可以用该计数器的值和目前网络的带宽比较

Page read/sec:每秒发出的物理数据库页读取数。这一统计信息显示的是在所有数据库间的物理页读取总数。由于物理 I/O 的开销大，可以通过使用更大的数据高速缓存、智能索引、更高效的查询或者改变数据库设计等方法，使开销减到最小。

Page write/sec: (写的页/秒) 每秒执行的物理数据库写的页数。

1. 判断应用程序的问题

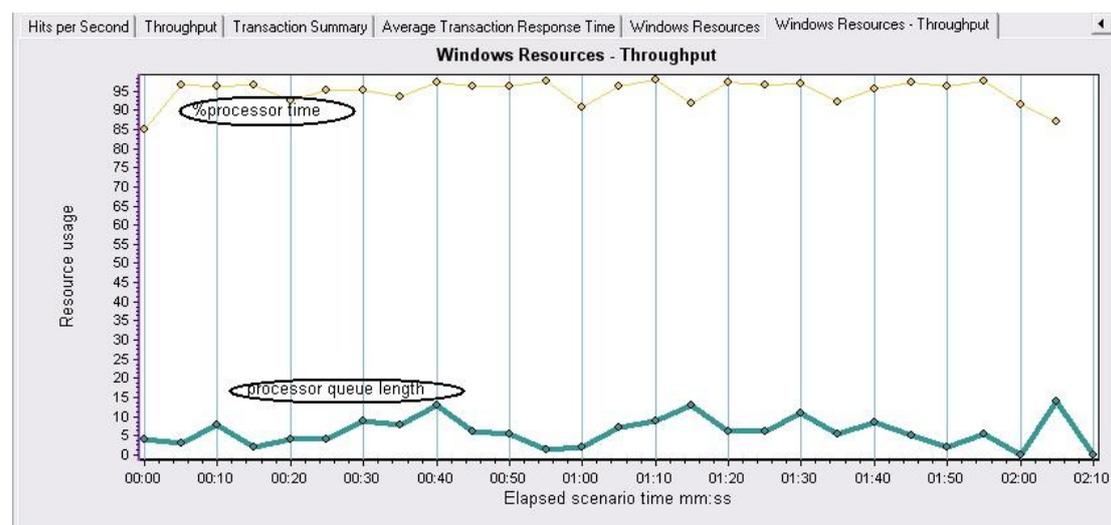
如果系统由于应用程序代码效率低下或者系统结构设计有缺陷而导致大量的上下文切换(context switches/sec显示的上下文切换次数太高)那么就会占用大量的系统资源,如果系统的吞吐量降低并且CPU的使用率很高,并且此现象发生时切换水平在15000以上,那么意味着上下文切换次数过高。



从图的整体看,context switches/sec变化不大,throughput曲线的斜率较高,并且此时的context switches/sec已经超过了15000.程序还是需要进一步优化。

2. 判断CPU瓶颈

如果processor queue length显示的队列长度保持不变(≥ 2)个并且处理器的利用率%Processor time超过90%,那么很可能存在处理器瓶颈.如果发现processor queue length显示的队列长度超过2,而处理器的利用率却一直很低,或许更应该去解决处理器阻塞问题,这里处理器一般不是瓶颈。

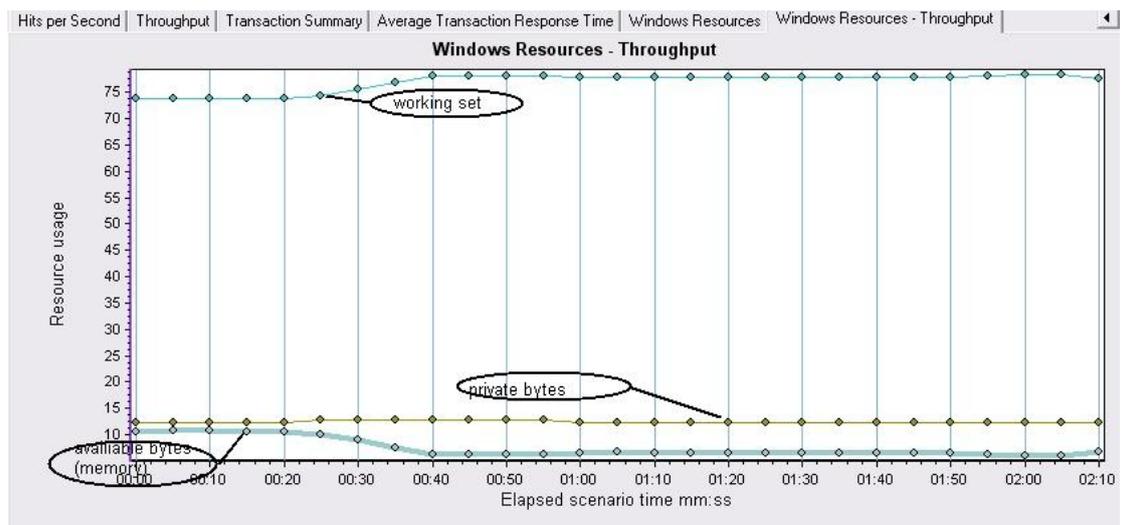


%processor time平均值大于95,processor queue length大于2.可以确定CPU瓶颈.此时的CPU已经不能满足程序需要.急需扩展。

3. 判断内存泄露问题

内存问题主要检查应用程序是否存在内存泄漏,如果发生了内存泄漏,process\private bytes计数器和process\working set 计数器的值往往会升高,同时available bytes的值会降低.内存泄漏

应该通过一个长时间的,用来研究分析所有内存都耗尽时,应用程序反应情况的测试来检验.



图中可以看到该程序并不存在内存泄露的问题.内存泄露问题经常出现在服务长时间运转的时候,由于部分程序对内存没有释放,而将内存慢慢耗尽.也是提醒大家对系统稳定性测试的关注.

附件:

CPU信息:

Processor\ % Processor Time 获得处理器使用情况。

也可以选择监视 Processor\ % User Time 和 % Privileged Time 以获得详细信息。

Server Work Queues\ Queue Length 计数器会显示出处理器瓶颈。队列长度持续大于 4 则表示可能出现处理器拥塞。

System\ Processor Queue Length 用于瓶颈检测

通过使用 Process\ % Processor Time 和 Process\ Working Set

Process\ % Processor Time过程的所有线程在每个处理器上的处理器时间总和。

硬盘信息:

Physical Disk\ % Disk Time

Physical Disk\ Avg.Disk Queue Length

例如, 包括 Page Reads/sec 和 % Disk Time 及 Avg.Disk Queue Length。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。但是, 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。

Physical Disk\ % Disk Time

Physical Disk\ Avg.Disk Queue Length

例如, 包括 Page Reads/sec 和 % Disk Time 及 Avg.Disk Queue Length。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。但是, 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。

请观察 Processor\ Interrupts/sec 计数器的值, 该计数器测量来自输入/输出 (I/O) 设备的服务请求的速度。如果此计数器的值明显增加, 而系统活动没有相应增加, 则表明存在硬件问题。

Physical Disk\ Disk Reads/sec and Disk Writes/sec

Physical Disk\ Current Disk Queue Length

Physical Disk\ % Disk Time

LogicalDisk\ % Free Space

测试磁盘性能时, 将性能数据记录到另一个磁盘或计算机, 以便这些数据不会干扰您正在测试的磁盘。

可能需要观察的附加计数器包括 Physical Disk\ Avg.Disk sec/Transfer、Avg.Disk Bytes/Transfer, 和 Disk Bytes/sec。

Avg.Disk sec/Transfer 计数器反映磁盘完成请求所用的时间。较高的值表明磁盘控制器由于失败而不断重试该磁盘。这些故障会增加平均磁盘传送时间。对于大多数磁盘, 较高的磁盘平均传送时间是大于 0.3 秒。

也可以查看 Avg.Disk Bytes/Transfer 的值。值大于 20 KB 表示该磁盘驱动器通常运行良好; 如果应用程序正在访问磁盘, 则会产生较低的值。例如, 随机访问磁盘的应用程序会增加平均 Disk sec/Transfer 时间, 因为随机传送需要增加搜索时间。

Disk Bytes/sec 提供磁盘系统的吞吐率。

决定工作负载的平衡

要平衡网络服务器上的负载, 需要了解服务器磁盘驱动器的繁忙程度。使用 Physical Disk\ % Disk Time 计数器, 该计数器显示驱动器活动时间的百分比。如果 % Disk Time 较高 (超过 90%), 请检查 Physical Disk\ Current Disk Queue Length 计数器以查看正在等待磁盘访问的系统请求数量。等待 I/O 请求的数量应当保持在不大于组成物理磁盘的主轴数的 1.5 到 2 倍。

尽管廉价磁盘冗余阵列 (RAID) 设备通常有多个主轴,大多数磁盘有一个主轴。硬件 RAID 设备在“系统监视器”中显示为一个物理磁盘;通过软件创建的 RAID 设备显示为多个驱动器(实例)。可以监视每个物理驱动器(而不是 RAID)的 Physical Disk 计数器,也可以使用 _Total 实例来监视所有计算机驱动器的数据。

使用 Current Disk Queue Length 和 % Disk Time 计数器来检测磁盘子系统的瓶颈。如果 Current Disk Queue Length 和 % Disk Time 的值始终较高,可以考虑升级磁盘驱动器或将某些文件移动到其他磁盘或服务器。

对于此处还要经过不断的学习和研究才能掌握得熟练.希望能够经常和大家一起学习交流,共同提高.由于本人也是初学者,在此献丑,有说明的不对的地方还希望专家指点.

2007.05.25

姜全尧