性能测试

中国软件评测中心

陈录萍

2004年7月

1

性能测试实践篇

中国软件评测中心

2004年7月

2

课程内容

测试需求分析
系统瓶颈分析
测试工具演示

中国软件评测中心

3

测试需求分析

测试管理的思路：

确定测试需求→计划测试→运行测试→跟踪缺陷

中国软件评测中心

4

测试需求分析

确定测试需求

任务	描述
定义测试范围	确定测试目的、测试对象与测试策略
创建测试需求	建立一个需求树去定义所有的测试需求
细化测试需求	对每一个测试需求题目产生一个细化的需求列表，详细描述需求、分配其优先级、必要的情况下添加附件
分析测试需求条款	通过报告或者图表的方式辅助分析测试需求，确保其符合测试范围

中国软件评测中心

5

测试需求分析

目标	回答问题
测量对最终用户的响应时间	要花多少时间做完一笔交易?
确定最优硬件配置	什么样的配置提供了最好的性能?
检查可靠性	系统能在无错情况下能承担多大及多长时间的负载?
检查软、硬件升级	这些升级对系统性能影响多大?
评估新产品	服务器应该选择哪些硬件与软件?
测试系统负载	在没有较大性能衰减的前提下，系统能够承受多大负载?
分析系统瓶颈	哪些因素降低交易响应时间

中国软件评测中心

6

测试需求分析

- 计划测试

任务	描述
定义测试策略	检查应用系统、系统环境以及测试资源以保证测试目标的实现
定义测试主题	将应用系统分为主题和最终实执行测试，创建测试计划将关键验证点及测试关元与主题
定义测试	为每一个主题确定测试类型，为测试计划树中的每个测试制定基础定义
创建测试需求范围	建立每个测试与测试需求的关系
设计测试步骤	为测试计划树中的手工测试添加步骤。测试步骤测试的操作、检查点以及期望的测试结果，确定哪些测试采用自动化测试方法
自动化测试	确定自动化测试中测试脚本如何生成，例如是用XXI自动测试工具，还是定制的或者第三方的测试工具
分析测试计划	通过报告或者图表的方式辅助分析测试计划数据，确保计划符合目标

测试需求分析

- 运行测试

任务	描述
创建测试组	定义测试组满足各种各样的测试目标。这些可能包括：例如
运行日程安排	制定测试执行的时间安排，并且给测试者分配任务
运行测试	在测试组中自动或者手工执行测试
分析测试结果	查看测试执行的结果，确认应用程序中缺陷是否被检测到。有图表和报告辅助分析。

系统瓶颈分析

最后的问题也是最难的问题！！！

- 具体问题具体分析
- 在总的指导思想下开展具体工作
- 分段排除法 很有效

性能测试概念

性能调优

- 查找系统瓶颈的根本原因
- 评估性能调整的效果
- 在测试环境下再现性能问题

性能测试概念

性能检测

- 在真实生产环境下，检测系统性能，评估并报告整个系统的性能和健壮情况
- 检查服务等级的满足情况
- 对系统的未来容量作出预测和规划

性能测试观点

观点1

仅仅建立最快的应用系统是不可取的，目标是达到最佳性能。

性能测试观点

观点2

把质量保证工作范围扩大到部署阶段之外，从而提高应用系统的质量。

性能测试观点

观点3

性能不仅仅是请求、点击次数和页面。

系统瓶颈分析举例

经验举例1

交易的响应时间如果很长，远远超过系统性能需求，表示耗费CPU的数据库操作，例如排序，执行aggregate functions（例如sum、min、max、count）等较多，可考虑是否有索引以及索引建立的是否合理；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。

系统瓶颈分析举例

经验举例2

测试工具可以模拟不同的虚拟用户来单独访问Web服务器、应用服务器和数据库服务器，这样，就可以在Web端测出的响应时间减去以上各个分段测出的时间就可以知道瓶颈在哪并着手调优。

系统瓶颈分析举例

经验举例3

UNIX资源监控（NT操作系统同理）中指标内存页交换速率（Paging rate），如果该值偶尔走高，表明当时有线程竞争内存。如果持续很高，则内存可能是瓶颈。也可能是内存访问命中率低。"Swap in rate"和"Swap out rate"也有类似的解释。

系统瓶颈分析举例

经验举例4

UNIX资源监控（NT操作系统同理）中指标CPU占用率（CPU utilization），如果该值持续超过95%，表明瓶颈是CPU。可以考虑增加一个处理器或换一个更快的处理器 。合理使用的范围在60%至70%。

系统瓶颈分析举例

经验举例5

UNIX资源监控（NT操作系统同理）中指标磁盘交换率（Disk rate），如果该参数值一直很高，表明I/O有问题。可考虑更换更快的硬盘系统。另外设置Tempdb in RAM，减低"max async IO"，"max lazy writer IO"等措施都会降低该值。

系统瓶颈分析举例

经验举例6

Tuxedo资源监控中指标队列中的字节数（Bytes on queue），队列长度应不超过磁盘数的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。

系统瓶颈分析举例

经验举例7

SQLServer资源监控中指标缓存点击率（Cache Hit Ratio），该值越高越好。如果持续低于80%，应考虑增加内存。 注意该参数值是从SQL Server启动后，就一直累加记数，所以运行经过一段时间后，该值将不能反映系统当前值。

优化调整设置

- CPU问题:
  - 考虑使用更高级的CPU代替目前的CPU
  - 对于多CPU，考虑CPU之间的负载分配
  - 考虑在其它体系上设计系统，例如增加前置机、设置并行服务器等。

优化调整设置

- 内存和高速缓存

内存的优化包括操作系统、数据库、应用程序的内存优化。

- 过多的分页与交换可能降低系统的性能
- 内存分配也是影响系统性能的主要原因
- 保证保留列表具有较大的邻接内存块
- 调整数据块缓冲区大小（用数据块的个数表示）是一个重要内容
- 将最频繁使用的数据保存在存储区中

优化调整设置

- 磁盘（I/O）资源问题

- 磁盘读写进度对数据库系统是至关重要的，数据库对象在物理设备上的合理分布能改善性能
- 磁盘镜像会减慢磁盘写的速度
- 通过把日志和数据库对象分布在独立的设备上可以提高系统的性能
- 把不同的数据库放在不同的硬盘上，可以提高读写速度。经常把数据库、回滚段、日志放在不同的设备上
- 把表放在一块硬盘上，把非簇的索引放在另一块硬盘上，保证物理读写更快

优化调整设置

- 调整配置参数
- 包括操作系统和数据库的参数配置
- 并行操作资源限制的参数（并发用户的数目、会话数）
- 影响资源开销的参数
- 与I/O有关的参数

优化调整设置

- 优化应用系统网络设置
- 可以通过数组接口来减少网络呼叫。不是一次提取一行，而是在单个往来往返中提取10行，这样做效率较高
- 调整会话数据单元的缓冲区大小
- 共享服务进程比专用服务进程提供较好的性能

举例说明

测试过程中出现的问题：

- 中间件客户端连接数
- 在配置文件中增大"连接数"值，保证客户端的连接数
- 队列阻塞问题
- 中间件的某个主要服务出现队列阻塞，在配置文件中增大相应队列中服务的数量
- 数据库优化选项问题
- 保证数据库操作均可以使用索引
- 负载均衡问题
- 大部分业务都集中在某一台服务器上，导致负载较重

举例说明

测试过程中出现的问题：

- 中间件客户端连接数
- 在配置文件中增大"连接数"值，保证客户端的连接数
- 队列阻塞问题
- 中间件的某个主要服务出现队列阻塞，在配置文件中增大相应队列中服务的数量
- 数据库优化选项问题
- 保证数据库操作均可以使用索引
- 负载均衡问题
- 大部分业务都集中在某一台服务器上，导致负载较重

数据库服务器典型性能问题

- 数据库服务器性能问题及原因分析
- 单一类型事务响应时间过长
  - 数据库服务器负载
  - 糟糕的数据库设计
  - 事务粒度过大
  - 批任务对普通用户性能的影响
- 并发处理能力差
- 锁冲突严重
  - 资源锁定造成的数据库事务超时
  - 数据库死锁

数据库服务器典型性能问题

数据库性能问题的一般解决办法

- 监视性能相关数据；
- 定位资源占用较大的事务并做出必要的优化或调整；
- 定位锁冲突，修改锁冲突发生严重的应用逻辑；
- 对规模较大的数据或者无法通过一般优化解决的锁冲突进行分布。

一个应用实例

- ××工程建设信息管理系统V1.0性能测试包括两次性能测试，第一次为性能检测与故障定位；第二次为针对第一次性能检测结果，进行调优之后的性能测试与评估。目前实施了第一次性能检测与故障定位。
- 第一次性能测试主要包括两个部分，即局域网测试和广域网测试。

中国软件评测中心 China Software Testing Center
31

局域网测试

- 测试目的:
  重点定位应用系统以及软、硬件支撑环境故障。
- 测试内容:
  在局域网测试环境下，对系统实施并发性能测试的同时，监控portal服务器（IBM Websphere Portal 4.2、DB2 7.2、IBM AIX 5L ML 4）和CM服务器（IBM Websphere 5.0、DB2 8.1、IBM AIX 5L ML 4）的资源使用情况。

中国软件评测中心 China Software Testing Center
32

局域网测试结论

1. 对比测试环境下的测试结果及故障定位
- 并发用户数未达到性能需求（100）。
- 业务交易响应时间较长。
- Portal处理速度慢，CM处理速度快。瓶颈在portal，CM处于空闲状态。
- portal CPU资源占用量大。
- 有失败SQL调用存在，造成客户端交易失败。

中国软件评测中心 China Software Testing Center
33

局域网测试结论

2. 真实业务测试环境下的测试结果及故障定位。
- 仍存在对比测试环境出现的所有问题。
- portal服务器内存在操作结束后不完全释放。

操作步骤	Portal 服务器内存占用
服务器重新启动	11.3%
启动 Domino	14.5%
启动应用系统	44.3%
压力测试结束时刻	81.7%
压力测试结束 15 小时之后	69.5%

中国软件评测中心 China Software Testing Center
34

局域网测试结论

- 首页portal的7个channel 加载期间CM 服务器CPU资源占用量过大，达到100％。
- 机房环境下交易的响应时间较长，机房环境与局域网环境对比，基本结论是：在局域网环境下交易的响应时间比在机房环境下交易的响应时间要长，差值较大的检查点如下。

案例	差值较大的检查点
制度文档	无
项目管理	登录、新增项目
工作记事	登录、新增工作记事、进入[illegible]

中国软件评测中心 China Software Testing Center
35

局域网测试结论

- 客户端交易失败时，数据库报错"连接失败"。数据库的最大连接数（目前设为40）和Websphere的最大连接数匹配关系在某种程度上导致该错误。
- 进程级的CPU与内存资源占用情况如下表，可以断定进行压力执行期间，Java进程在争用资源。
- 网络测试环境下必须在承受的并发用户数及交易响应时间之间取得平衡。

中国软件评测中心 China Software Testing Center
36

局域网测试结论

案例			CPU%（top4）				Memory%（top4）物理内存/虚拟内存			
			1	2	3	4	1	2	3	4
工作记事		进程	Java	Java	Java	httpd	Java	Java	Java	Java
		数值	45	5	3	0.8	14/29	12/24	5/11	1/3
项目管理		进程	Java	Java	Java	server	Java	Java	Java	Java
		数值	80	7	5	0.8	17/26	15/24	5/9	2/3
制度文档	信息上传	进程	Java	Java	Java	Db2agent	Java	Java	Java	Java
		数值	40	5	3	2	19/26	17/24	5/8	1/2
	文件上传与下载	进程	Java	Java	Java	server	Java	Java	Java	Java
		数值	34	30	3	0.8	21/28	18/25	5/8	1/2

中国软件评测中心 China Software Testing Center

37

局域网测试结论

3. 优化重点考虑

- 优化重点考虑portal服务器。
- portal与CM的功能划分，即EJB的业务逻辑划分是否合理。
- CM、Portal、websphere、DB2的调优。
- 源代码在开发过程中的优化，例如是否考虑连接缓冲池、动态和静态请求调用是否合理等。
- CM的调优从应用逻辑、CM服务器性能两方面入手。
- 将操作之间的关联度降低。
- 数据库的最大连接数以及Websphere的数匹配关系应该重点考虑。

中国软件评测中心 China Software Testing Center

38

局域网测试结论

- 调优步骤应该为：
  - Ø 系统软硬件资源的调优。
  - Ø 首页portal。
  - Ø 响应较慢的网页组件。
  - Ø 影响并发用户数的链接。
  - Ø 系统硬件资源。

中国软件评测中心 China Software Testing Center

39

广域网测试

- 测试目的：
  重点测试网络环境对应用系统的影响，定位网络故障。
- 测试内容：
  在广域网环境下，对系统实施并发性能测试的同时，监控网络资源、同时关注portal服务器及CM服务器资源使用情况。重点测试网络环境对应用系统的影响，定位网络故障。

中国软件评测中心 China Software Testing Center

40

广域网测试结论

1. 交易响应时间对比

对比局域网和广域网交易平均响应时间，同一交易在相同带宽下，广域网测试环境下的响应时间，都比局域网测试环境下的响应时间长，响应时间差见下表。

案例 / 响应时间差	登录 (init_Transaction)	交易1 (Action1_Transaction)	交易2 (Action1_Transaction)
不限带宽	275.202 s	216.775 s	509.822 s
2M带宽	14.672 s	62.051 s	251.035 s
512K带宽	23.577 s	20.774 s	156.159 s

中国软件评测中心 China Software Testing Center

41

广域网测试结论

2. 应用网络故障定位

1)广域网环境中网络传输时间所占比例较高，Potal服务器发出的数据在广域网络上的传输时间相对服务器本身处理时间要长。网络带宽的影响和其他应用程序争用带宽的影响不是主导因素。因而重点考察Potal服务器的网络发送时间。

2)考查Potal服务器发出的数据帧的明细，可见应用数据在广域网上存在很多帧丢失和TCP重传的现象。（Error: Frame only seen on sending side. prossibly lost by network; TCP Retransmission of earlier frame xxxx）。通过分析数据中的TCP ACK帧，可知TCP重传不是由客户端接收缓存满而不能及时接收数据造成的。要综合考虑网络品质问题和数据发送端的有关TCP重传参数设置。

中国软件评测中心 China Software Testing Center

42

广域网测试结论

3)例如针对起始序号784、结尾序号973的数据帧进行包跟踪分析，序号784的数据丢失，导致多次重传，序号791、819、855、973均为该数据帧的重传帧，且重传的延时不断递增（重传开始时间间隔为1.407542578sec、3.000778311sec、6.001580091 sec）。因此造成应用数据传输时间增加。

4)分析出错重传的数据帧，发现字节少的帧（如60字节的TCP ACK）较少出现丢失重传，字节多（如1514字节的HTTP[HTTP/1.1 200 OK]）丢失重传较多。

5)数据传输的包大则增加了帧的传输次数，同时增加了出错概率。

广域网测试结论

3. 应用网络性能分析

1)应用程序会话分析

- 针对不同的案例，Portal服务器到CM服务器会话往返行程基本保持一致，可以看到系统应用通讯模式基本一致，有利于系统维护。
- 加压机到Portal服务器运行在广域网环境下，往返行程次数越多，受网络品质、网络延迟影响越大。例如"项目管理"案例达到138次。
- 建议降低加压机到Portal服务器的往返行程次数。

广域网测试结论

- 本案例中不存在相同的请求再次发出的现象。
- 线程与会话（bounce）以及包（packet）之间相关联，包之间的相对时间（relative time）决定了包传输的效率。
- 建议对关键线程进行相关调优，有必要参考包数据。

广域网测试结论

2)应用程序线程分析（以"项目管理"案例为例）

- 应用往返行程（App. Turns）最大的三个线程受网络品质、网络延迟影响最大。
- 分析往返行程最大的三个线程主要性能数据，其中线程"HTTP:[GET /wps/portal HTTP/1.1]"与"TCP: 54760<->50000"持续时间最长，并且线程"TCP: 54760<->50000"发出和接收的字节数和帧数最大。
- 线程之间系统闲歇（gap）达到3秒，可见系统空闲时间在控制范围之内，基本可以接收。
- 线程的请求与回应基本在1秒钟之内，但也有约3秒钟的响应时间。

广域网测试结论

3)交易峰值分析（以"项目管理"案例为例）

- 交易最高峰值达到2.4M,1M以下的峰值分布较均匀。
- 交易最高峰值2.4M所对应的执行线程是调优的关键。
- 建议降低交易最高峰值,调整这一时刻某些线程执行的起始时间。

4)帧数据分析

- 小于100字节和大于1024字节的帧在传输的帧中占的比例最高。

5)模拟不同带宽（例如10M、2M、512k、56k）、延迟（例如150ms、100ms、50ms、1ms）、负载（例如10%、50%）等网络应用情况，进行应用响应时间预测。

广域网测试结论

4. 网络品质监控

1)网络流量分析

- 从不同带宽的数据对比来看，随着网络带宽的不断减小，进行同样交易所传输的数据量有较大的增长趋势，这种现象说明随着网络带宽的下降，网络重传现象不断加重，这与前面应用网络故障定位的结论一致。

2)应用分布分析

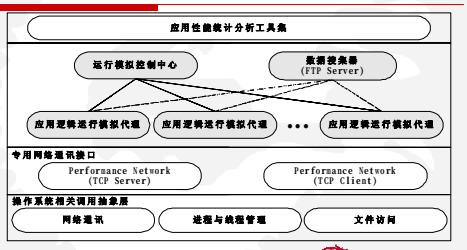
- 模拟不同带宽下的带宽利用率，可以看出：当带宽为10M时带宽利用率最大值接近15%，带宽为2M时带宽利用率最大值接近50%，带宽为512K时带宽利用率最大值接近200%（注：模拟带宽为独占带宽）。

广域网测试结论

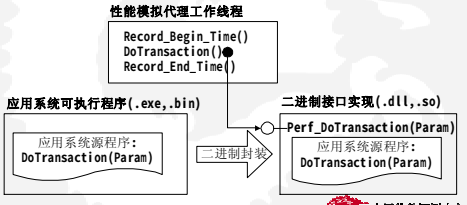
3)防火墙延迟分析

- 对比不同带宽下监测到的防火墙延迟，可以看到延迟值较稳定，基本保持在14ms左右。
- 通过观察相同时间段两个探针监测到的应用数据，发现数据量相差很小，这说明防火墙的丢包率不是很大，结合前面的分析可以看出，丢包现象主要发生在广域网传输段。

开发测试工具



开发测试工具



开发测试工具

接口名称	功能	接口函数原型(C语言描述)
事务处理逻辑类型	此接口提供应用系统相关的事务类型ID，通常情况下应用系统由多种事务组成，评测系统应该能够区分它们	Int Perf_GetTransactionID(void);
接口初始化	初始化应用系统，诸如创建数据库连接等操作在此接口中进行	int Perf_InitLib(void);
执行事务处理	实际事务处理接口，事务执行的起止时间和执行状态在此记录	int Perf_DoTransaction(void* pBuffer,int nBufferSize);
接口资源释放	释放应用系统资源，诸如关闭数据库连接等操作在此函数	int Perf_FreeLib(void);

开发测试工具

操作系统	开发环境
Win32(Windows NT4/2000/XP)	Visual C++ 6.0 SP5
SUN Solaris8(x86)	Forte C++ 6 Update 2
Linux(x86 kernel 2.4.7 glibc 2.2.4)	KDevelop 2.0

实践篇总结

测试需求分析
系统瓶颈分析
测试工具演示