

为什么要做播放器的性能测试呢？

产品名称	为什么要做播放器的性能测试呢？
公司名称	超越检测技术（深圳）有限公司
价格	500.00/件
规格参数	
公司地址	深圳市宝安区燕罗街道洪桥头社区兆福达工业区综合楼B栋一单元502检测实验室
联系电话	18138236659 18138236659

产品详情

对于互联网应用软件，性能

是其质量的一个非常重要的组成部分。作为解决[软件性能](#)

问题的重要手段，软件性能测试已经广为人们所熟悉，并受到很高的关注。一般而言，软件性能测试都是在项目的后期才开展，被测试的对象通常是已经具有一定稳定性的产品。而实际上，软件性能测试应贯穿于整个[软件生命周期](#)中，和[功能测试](#)一样，软件性能测试也分为几个阶段。

软件生命周期与软件性能测试

不论哪种[软件生命周期模型](#)，[需求分析](#)

、设计、编码、测试和运行维护这几个阶段都是其中的基本要素，只是在不同的软件生命周期模型中可能迭代、合并、拆分或重组这几个阶段，在此不做过多的描述。与其他几个阶段相对应，测试从软件开发过程按阶段可以划分为：[单元测试](#)、[集成测试](#)、[系统测试](#)

，在其他的书上可能还能见到诸如[确认测试](#)

、验收测试等名词，但是前3种测试确实是最基本的测试活动，而其他的测试活动只是在某些软件开发过程中会发生。

值得注意

的是，通常在谈论

单元测试、集成测试和系统测试时，

其实仅仅谈论的是不同阶段的[功能测试](#)；而当讨论[软件性能](#)

测试时，绝大多数的情况是，一个已经开发完毕或基本开发完毕的软件，测试人员用一种或几种软件性能测试工具，以尽量模拟真实用户行为的方式对该软件进行并发操作，收集并比较不同场景的结果，然后对软件的性能进行分析，这个活动通常发生在系统测试阶段，甚至更往后的阶段，如运行维护阶段。

一直以来，软件性能测试跟[单元测试](#)、[集成测试](#)

似乎都是绝缘的。可是它们真的应该是绝缘的吗？没有任何理由可以说明软件性能测试跟单元测试、集成测试无关，除非你认为“这太难了，我不会做”。

[编辑本段](#)测试术语响应时间

我把“响应时

间”的概念确定为“对请求

作出响应所需要的时间”，把响应时间作为用户视

角的[软件性能](#)的主要体现。响应时间划分为“呈现时间”和“系统响应时间”两个部分。

其中

“呈现时

间”取决于数据在

被客户端收到响应数据后呈现页面

所消耗的时间、而“响应时间”指J2EE[应用服务器](#)

从请求发出开始到客户端接受到数据所消耗的时间。软件性能测试一般不关注“呈现时间”，因为呈现时间很大程度上取决于客户端的表现。在这里我们没有使用很多软件性能测试定义中的概念--“系统响应时间”定义为“应用系统从请求发出开始到客户

端接收到最后一个[字节](#)

数据所消耗的时间”，没有使用这种标准的原因是，可以使用一些编程技巧在数据尚未完全接收完成时进

行呈现来减少用户感受到的响应时间，对于HNDLZCGLXT的这个项目中，我们针对C/S系统采用前者标准，对于B/S我们依然采用后一种标准。

并发用户数

我把"[并发用户数](#)"与"[同时在线数](#)

"进行区别对待，我的"并发用户数"的标准是:并发用户数取决于测试对象的目标业务场景，因此，在确定这个"并发用户数"前，必须(必要)先对用户的业务进行分解、分析出典型的业务场景(也就是用户最常使用、最关注的业务操作)，然后基于场景采用某些方法(有多种计算并发用户数的数学模型与公式)获得"并发用户数"。

这样做的原因是:假设一个应用系统、最高峰有500人同时在线、但这500人却不是并发用户数、因为假设在一个时间点上、有50%的人在填写复杂的表格(填写表格动作对服务器没有任何负担、只有在"提交"动作的时候才会对[服务器系统](#)

构成压力)、有40%的人在不停的从一个页面跳转到另外一个页面(不停发出请求与回应、产生服务器压力)、还有10%的人挂在线上，没有任何操作在发呆:)(没有对服务器构成压力的动作)。因此只有那40%的人真正对服务器产生了压力，从这里例子可以看出、[并发用户数](#)

关心的是

不但是业务并发用

户数、还取决于业务逻辑、业务场景。因此我们需要本文第六部分[软件性能](#)测试文档4、5、6。

吞吐量

我把吞吐量定义为"单位时间内系统处理的客户请求的数量"，直接体现软件系统的性能承载能力，对于交互式应用系统来说、吞吐量反映的是服务器承受的压力、在容量规划的测试中、吞吐量是一个重要指标、它不但反映在[中间件](#)、数据库上、更加体现在硬件上。我们在以下方面利用这个指标:

(1)

用来协助设计[软件性能](#)

测试场景，衡量软件性

能测试是否达到了预计的设计目标、比如J2EE

应用系统的[连接池](#)、数据库事务发生频率、事务发生次数。

(2) 用来协助分析性能瓶颈、参照本文第二部分总的RBI方法。

性能计数器

性能计

数器式描述服

务器或操作系统性能的一些

数据指标、例如对WINDOWS来说使用内存数、[CPU使用率](#)、进程时间等都是常见的计数器。 [Page]

对于性能计数器这个指标来说、需要考虑到的不但有硬件计数器、web服务器计数器、Weblogic服务器计数器、Servlet性能计数器、EJB2的性能计数器、JSF性能计数器、JMS性能计数器。找到这些指标是使用性能计数器的第一步、关键是找到性能瓶颈、确定系统阈值、提供优化建议才是性能计数器使用的关键。性能计数器复杂而繁多、与代码上下文环境、系统配置情况、系统架构、开发方式、使用到的规范实现、工具、类库版本都有紧密的联系、在此不作赘述。

思考时间

我把思考时间确定为"休眠时间"。从业务系统的角度来说，这个时间指的是用户在惊醒操作时、每个请求之间的时间间隔、从[自动化测试](#)

的角度来说、要真实的测试模拟用户操作、就必须

在[测试脚本](#)

中让各个操作之间等待一段时间、体现在脚本上就是在操作之间放置一个Think的函数，体现为脚本中两个请求语句之间的间隔时间、不同的测试工具提供了不同的函数或方法来实现思考时间、比如HP LoadRunner和IBM Rational Performance Tester的方式就完全不同。