校内音乐分享站

测试报告

# 1.1 校内音乐分享站系统概述

## 1.1.1系统功能

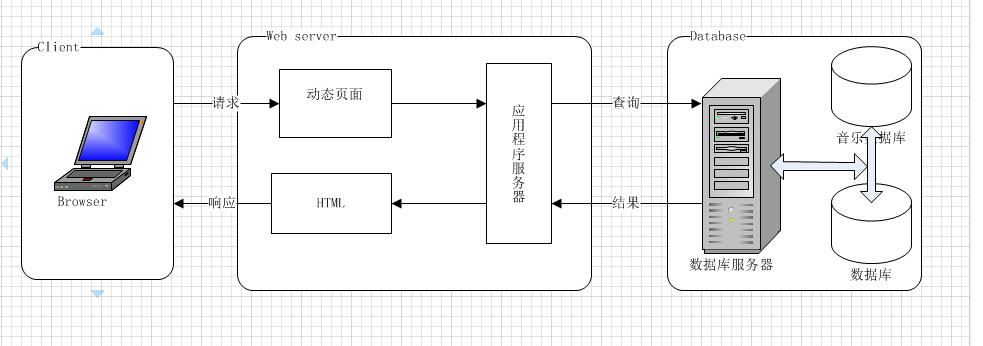
本次要进行测试的是一个校内音乐分享站系统。系统包括以下功能模块：

1. 上传音乐：上传本地音乐至网站
2. 音乐博客：在页面添加喜欢的MV、音乐，以及对该音乐的理解，听音乐时的感慨等等。其他人可见。
3. 点评：分享用户翻唱与点评，分享原创音乐
4. 播放功能：将音乐按流派分类，提供自动播放、选歌的功能，并可生成播放列表将指定歌曲按顺序播放。
5. 音乐搜索：提供按歌手搜索、按歌名搜索等功能，并默认将歌曲按照歌手分类。
6. 收藏音乐：将音乐收藏进“我最喜欢”列表，并可由此列表生成一个播放列表。收藏列表默认所有人可见。
7. 热门榜：根据用户每周收藏次数/下载次数将歌曲按从大到小排序，列出一个热门榜单，收藏数最多歌曲即为最热门。所有人可见。
8. 歌曲下载：允许用户将音乐下载至本地。
9. 多平台同步：收藏的歌曲自动同步到手机上面，支持Android和iOS。
10. 推荐歌曲：提供URL链接，让用户可以将喜欢的音乐分享给其他平台用户。
11. 截取音乐：用户可以自由截取一首音乐的部分，并下载至本地。

## 1.1.2系统的体系结构

校内音乐分享站系统为典型的B/S结构。客户端都是通过浏览器访问应用系统。Web服务器为Apache，数据库为MYSQL。浏览器和Web服务器之间基于HTTP协议。

**系统的体系结构如下图：**



## 1.1.3开发环境

本校内音乐分享站系统开发的软件环境如下：

1. 操作系统：windows7
2. Web服务器：Apache
3. 数据库：MYSQL
4. 开发语言和工具：php+javascript + html5 + eclipse
5. 浏览器：google浏览器

# 1.2 校内音乐分享站系统测试计划

## 1.2.1测试需求

此系统属于一般类型的web应用，用户要求各功能正常使用，系统响应比较快，运行稳健，能满足500人正常使用。此系统用户可以建立个人音乐博客（在页面添加喜欢的MV、音乐，以及对该音乐的理解，听音乐时的感慨等等），歌曲下载，上传音乐，音乐搜索。

此次测试的目的就是检查核心模块功能是否正常，验证系统性能是否满足应用需求。这次测试的具体内容如下：

### 1、功能测试

在功能测试中，重点测试的是登录模块、上传音乐模块。

### 2、性能测试

在性能测试中，需要测试不同负载下系统的表现，并获得系统的响应能力、负载能力、吞吐量和资源利用率等性能指标。

### 3、可用性测试

### 4、兼容性测试

**5**、安全性测试

## 1.2.2测试资源

### 1、人力资源

|  |  |  |
| --- | --- | --- |
| 姓名 | 角色 | 职责 |
| 袁泉 | 测试人员 | 设计并执行功能和性能测试 |

### 2、测试环境

1. 硬件设备：1台PC机
2. 软件环境。

操作系统：windows 7

WEB服务器：google

1. 网络环境：学校内部的以太网

### 3、测试工具

|  |  |
| --- | --- |
| 用途 | 工具 |
| 功能测试 | WinRunner |
| 性能测试 | LoadRunner |
| 安全性测试 | IBM Rational Appscan |

### 4、文档资料

《应用需求规格说明》

《项目建议书》

## 1.2.3测试策略

### 1、功能测试

功能测试的目的是保证系统的功能正常，例如数据输入是否正确。对交互的输出或结果进行分析，以核实应用程序的功能。

重点测试的是登录模块、上传音乐模块，测试策略见下表：

**登录模块测试策略**

|  |  |
| --- | --- |
| 测试策略项 | 登录模块测试 |
| 测试类型 | 功能测试 |
| 测试技术 | 手工测试 |
| 测试通过/失败标准 | 95%测试用例通过，并且所有缺陷全部解决 |
| 特殊考虑 | 无 |

**上传音乐模块测试策略**

|  |  |
| --- | --- |
| 测试策略项 | 上传音乐模块测试 |
| 测试类型 | 功能测试 |
| 测试技术 | 手工测试 |
| 测试通过/失败标准 | 95%测试用例通过，并且所有缺陷全部解决 |
| 特殊考虑 | 需要进行表单测试和数据库测试 |

### 2、性能测试

性能测试主要是对响应时间、事务处理速率和其他与时间有关的需求进行评测和核实监测系统性能是否都以满足。

在本次的性能测试中，主要是进行并发性能测试。

本次测试的是登录模块、音乐搜索模块、上传音乐模块，测试策略见下表：

**登录模块并发性能测试**

|  |  |
| --- | --- |
| 测试策略项 | 登录模块并发测试 |
| 测试技术 | 采用LoadRunner测试工具自动化测试 |
| 测试通过/失败标准 | 80%的事务平均响应时间不超过8s,每一事务的响应时间不超过12s |
| 特殊考虑 | 1. 可创建“虚拟的”用户负载来模拟多个（通常为数百个）客户机 2. 最好使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载 3. 多用户不同网络条件下的连接速度是否满足要求 |

上传音乐**并发性能测试**

|  |  |
| --- | --- |
| 测试策略项 | 上传音乐模块并发测试 |
| 测试技术 | 采用LoadRunner测试工具自动化测试 |
| 测试通过/失败标准 | 80%的事务平均响应时间不超过10s，每一事务的响应时间不超过12s |
| 特殊考虑 | （1）可创建“虚拟的”用户负载来模拟许多个（通常为数百个）客户机  （2）最好使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载  （3）对用户提交的表单进行检查 |

### 3、用户界面测试

|  |  |
| --- | --- |
| 检查项 | 测试人员的类别及其评价 |
| 窗口切换、移动、改变大小是否正常 | Yes |
| 各种界面元素的文字是否正确（如标题、提示等） | Yes |
| 各种界面元素的状态是否正确（如有效、无效、选中等状态） | Yes |
| 各种界面元素是否支持键盘操作 | Yes |
| 各种界面元素是否支持鼠标操作 | Yes |
| 数据项能否正常回显 | Yes |
| 对于常用的功能，用户能否不必阅读手册就能使用 | Yes |
| 执行有风险的操作时，是否有“确认”、“放弃”等提示 | Yes |
| 操作顺序是否合理 | Yes |
| 按钮排序是否合理 | Yes |
| 导航帮助是否明确 | Yes |
| 提示信息是否规范 | Yes |
| 在不同的浏览器下用户界面的所有元素是否正常 | Yes |
| 在同一种浏览器下，浏览器的版本不同用户界面是否正常 | Yes |

### 4、安全性测试

手动测试

### 5、兼容性测试

略

## 1.2.4测试标准

### 1、测试特性

功能性（50%）、效率（35%）、可靠性（5%）、可维护性（10%）

### 2、输出准则

1. 文档：系统测试说明、系统测试报告。
2. 覆盖率：计划测试覆盖率100%，执行测试覆盖率99%。
3. 功能质量目标：缺陷遗留数：严重缺陷0个，较严重缺陷1个，一般缺陷3个，次要缺陷7个。
4. 性能质量目标：

### 3、缺陷严重级别定义

严重缺陷：不能执行正常工作功能或重要功能，是系统崩溃或资源严重不足。例如由程序引起的死机。

较严重缺陷：严重的影响系统要求或基本功能的实现，且没办法更正。例如功能与需求不符、程序接口错误等。

一般错误：严重地影响系统要求或基本功能的实现，但存在合理的更正办法。例如界面错误，输出内容及打印格式错误等。

次要错误：使操作者不方便或遇到麻烦，但不影响执行工作或功能实现。例如辅助说明描述不清楚，显示格式不规范等。

# 1.3 校内音乐分享站系统功能测试

## 1.3.1登陆模块的测试

下面对主页面上的登录模块进行功能测试。

### 测试用例设计

根据登录操作的特点，采用边界值方法和等价类设计测试用例。登录个人主页的测试用例如下所示：

**登录校内音乐分享站测试用例**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目名称 | 登录校内音乐分享站主页测试 | 项目编号 |  | | | | | |
| 开发人员 |  | 模块名称 | 登录校内音乐分享站个人主页 | | | | | |
| 用例作者 | 袁泉 | 参考信息 | 需求规格说明书、设计说明书 | | | | | |
| 测试类型 | 功能测试 | 设计日期 | 2014-06-28 | | 测试人员 | | 袁泉 | |
| 测试方法 | 手工测试 | 测试日期 |  | |  | |  | |
| 测试对象 | 测试用户能否正常登录 | | | | | | | |
| 前置条件 | 存在正确的用户名和密码；登录页面正常装载； | | | | | | | |
| 用例编号 | 操作 | 输入数据 | | 预期结果 | | 实际结果 | | 测试状态（P/F） |
| 01 | 输入正确的帐号、正确的密码，点击“登录” | 帐号：XXX  密码：XXX | | 正常登录 | | 正常登录，转入对应的系统页面 | | P |
| 02 | 输入正确的帐号、正确的密码，按ENTER键 | 帐号：XXX  密码：XXX | | 正常登录 | | 正常登录，转入对应的系统页面 | | P |
| 03 | 输入错误的帐号或者没有注册的帐号，点击“登录” | 帐号：2443095540  密码：123456abc | | 不能正常登录，显示“帐号不存在或者错误”的提示 | | 不能正常登录，提示“您输入的帐号或者密码不正确，请重新输入。” | | F |
| 04 | 首次输入正确的帐号、错误的密码，点击“登录” | 帐号：2443095549  密码：123456 | | 不能正常登录，显示“密码错误” | | 不能正常登录，提示“您输入的帐号或者密码不正确，请重新输入。” | | F |
| 05 | 第二次输入正确的帐号、错误的密码，点击“登录” | 帐号：2443095549  密码：123 | | 不能正常登录，显示“密码错误” | | 不能正常登录，提示“您输入的帐号或者密码不正确，请重新输入。” | | F |
| 06 | 第三次输入正确的帐号、错误的密码，点击“登录” | 帐号：2443095549  密码：1 | | 不能正常登录，显示“密码错误” | | 不能正常登录，提示“您今天修改过密码，请使用新的密码登录”，并在登录界面上出现验证码 | | F |
| 07 | 第四次输入正确的帐号、错误的密码和正确的验证码，点击“登录” | 帐号：2443095549  密码：12  验证码：图片中的字母 | | 不能正常登录，显示“密码错误” | | 不能正常登录，提示“您输入的帐号或者密码不正确，请重新输入。” | | F |
| 08 | 输入正确的帐号、正确的密码和正确的验证码，点击“登录” | 帐号：2443095549  密码：123456  验证码：图片中的字母 | | 正常登录 | | 正常登录，转入对应的系统页面 | | P |
| 09 | 输入正确的帐号、正确的密码和错误的验证码，点击“登录” | 帐号：2443095549  密码：123456  验证码：输入的字母与图片中的字母不一致（字母个数相同） | | 不能正常登录，显示“验证码错误” | | 不能正常登录，提示“您输入的验证码不正确，请重新输入。” | | F |
| 10 | 输入正确的帐号、正确的密码和错误的验证码，点击“登录” | 帐号：2443095549  密码：123456  验证码：输入的字母与图片中的字母不一致（个数少） | | 不能正常登录，显示“验证码错误” | | 不能正常登录，提示“请输入完整的验证码！” | | F |
| 11 | 输入正确的帐号、密码为空，点击“登录”（此时没有验证码） | 帐号：2443095549  密码： | | 不能正常登录，显示“密码不能为空” | | 不能正常登录，提示“您还没有输入密码！” | | F |
| 12 | 帐号、密码都为空，点击“登录” | 帐号：  密码： | | 不能正常登录，显示“帐号为空” | | 不能正常登录，提示“您还没有输入帐号！” | | F |
| 13 | 帐号为空，只输入密码 | 帐号：  密码：123456abc | | 不能正常登录，显示“帐号为空” | | 不能正常登录，提示“您还没有输入帐号！” | | F |
| 14 | 帐号正确，但其后有1至多个空格，密码正确 | 帐号：2443095549+2空格  密码：123456abc | | 不能正常登录，显示“帐号错误” | | 正常登录，转入对应的系统页面 | | P |
| 15 | 用户名和密码正确，但密码后加1个至多个空格 | 帐号：2443095549  密码：123456abc+2空格 | | 不能正常登录，显示“密码错误” | | 不能正常登录，提示“您输入的验证码不正确，请重新输入。” | | F |
| 16 | 使用Tab键 | 光标在帐号框内，按tab键2次 | | 光标可依次移动到密码输入框和页面上的图片 | | Tab键功能正常使用 | | P |
| 17 | 在文本输入框中按BackSpace键 | 在帐号框输入：2443095549，光标在字符串末尾，按4次BackSpace键 | | 帐号本文框显示：244309 | | BackSpace键能正常使用 | | P |
| 18 | 在文本输入框中使用左右箭头 | 在帐号输入框中使用左右箭头 | | 光标必须能跟踪到相应位置 | | 左右箭头能正常使用 | | P |
| 19 | 在文本输入框中使用delete键 | 在帐号输入框中使用delete键 | | 能正常删除 | | Delete键能正常使用 | | P |
| 20 | 在文本输入框中点击鼠标 | 在帐号输入框内单击鼠标 | | 光标必须能跟踪到相应位置 | | 正常 | | P |
| 21 | 在文本输入框中双击鼠标 | 在帐号输入框内双击鼠标 | | 输入框中文本被选中 | | 输入框中文本被选中 | | P |
| 22 | 输入帐号，等待较长时间才输入密码 | 帐号：2443095549  等待五分钟输入密码  密码：123456abc | | 正常登录 | | 正常登录，转入对应的系统页面 | | P |
| 23 | 输入帐号，马上切换到其他程序，过一段时间再切换回来 | 帐号：2443095549  切换到word程序，过一分钟在切换回来 | | 光标位置应停在原处 | | 光标位置应停在原处 | | P |

### 2、执行测试

直接按照测试用例的要求，输入测试数据，观察运行的结果与预期结果的异同，以判断测试是否通过。

### 3、测试结果

通过手动测试，发现了1个轻微的缺陷，是：

（1）输入用户名和密码错误三次后，系统不会锁住用户

## 1.3.2上传音乐模块测试

此模块包括两个子模块，上传音乐和创建专辑，主要采用场景测试法进行测试。

### 1、测试用例设计

使用场景法进行测试时必须首先分析出对象及基本事务流和备选事务流。

（1）上传音乐模块开始是进入音乐管理。上传音乐包括6个基本步骤：

* 1. 点击“上传音乐”
  2. 点击“添加音乐”
  3. 选择音乐
  4. 选择音乐专辑
  5. 点击“上传音乐”
  6. 点击“完成”

这五步依次正确操作便形成了基本事务流。

在该模块中有两条备选事务流，分别是：

1. 备选流一：在基本流4》中，添加音乐专辑。有两种情况：一是未创建音乐专辑，无音乐专辑可选，需要添加音乐专辑；二是已有音乐专辑，准备另外添加新音乐专辑。
2. 备选流二：在基本流5》中，网络速度太慢，导致传输错误。

从基本流开始，将基本流和备选流结合起来，可以确定如下用例场景：

场景1：基本流

场景2：基本流，备选流一

场景3：基本流，备选流二

由场景生成测试用例是通过确定某个特定条件来完成的，这个特定条件将导致特定用例场景的执行。根据上面的场景设计测试用例，如下表：

**上传音乐测试用例**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目名称 | 上传音乐测试 | 项目编号 | |  | | | | |
| 开发人员 |  | 模块名称 | | 上传音乐模块测试 | | | | |
| 用例作者 | 袁泉 | 参考信息 | | 需求规格说明书、概要设计说明书 | | | | |
| 测试类型 | 功能测试 | 设计日期 | | 2014年6月28日 | | 测试人员 | | 袁泉 |
| 测试方法 | 手工（黑盒测试） | 测试日期 | | 2014年6月28日 | | | | |
| 测试对象 | 上传音乐功能 | | | | | | | |
| 前置条件 | 用户进入音乐管理页面，专辑中已有“流行”的专辑。 | | | | | | | |
| 用例编号 | 场景 | 输入数据 | | | 预期结果 | | 实际结果 | |
| 专辑 | 监测网速 | |
| 01 | 场景一：上传音乐成功 | 选择“流行”专辑 | 网速很快 | | 页面提示：上传完成，是否继续 | | 页面提示：上传完成，是否继续 | |
| 02 | 场景二：添加专辑（测试未创建专辑情况） | 无专辑可选 | 网速很快 | | 页面提示：未创建专辑，转入创建专辑页面 | | 页面提示：未创建专辑，转入创建专辑页面 | |
| 03 | 场景二：添加专辑（已有专辑，添加新专辑） | 点击“选择专辑” | 网络很快 | | 转入选择的专辑页面 | | 转入选择的专辑页面 | |
| 04 | 场景三：网络传输速度慢 | 选择“流行”专辑 | 网络速度0—1kb | | 页面提示：网速慢，请稍后重传 | | 页面提示：网速慢，请重传 | |

(2) 创建音乐专辑模块开始是进入音乐管理。创建音乐专辑模块包括7个基本步骤：

1》 点击“创建专辑”

2》 填写专辑名称

3》 填写专辑描述

4》 选择分类

5》点击确定

这五步依次正确操作便形成了基本事务流。

在该模块中有一条备选事务流，其是：

1. 备选流一：在基本流2》中，专辑名称为空。

由场景生成测试用例是通过确定某个特定条件来完成的，这个特定条件将导致特定用例场景的执行。根据上面的场景设计测试用例，如下表：

**创建专辑测试用例**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 项目名称 | 创建专辑测试 | 项目编号 |  | | | | |
| 开发人员 |  | 模块名称 | 创建音乐搜索模块 | | | | |
| 用例作者 | 袁泉 | 参考信息 | 需求规格说明书、概要设计说明书 | | | | |
| 测试类型 | 功能测试 | 设计日期 | 2014年6月28日 | | 测试人员 | | 袁泉 |
| 测试方法 | 手工（黑盒测试） | 测试日期 | 2014年6月28日 | | | | |
| 测试对象 | 创建专辑功能 | | | | | | |
| 前置条件 | 用户进入专辑管理页面，专辑中已有“流行”的专辑。 | | | | | | |
| 用例编号 | 场景 | 输入数据 | | 预期结果 | | 实际结果 | |
| 专辑名称 | |
| 01 | 场景一：创建专辑成功 | 填写专辑名称“古典” | | 页面提示：创建完成，是否传音乐 | | 页面提示：创建完成，是否上传音乐 | |
| 02 | 场景二：创建专辑不成功 | 未填写专辑名称 | | 页面提示：请填写专辑名称 | | 页面提示：请填写专辑名称 | |

### 2、执行测试

该模块采用手动方式执行测试。根据各个测试用例的说明，一次执行每个测试用例，得出测试结果。

### 3、测试结果

执行每个测试用例之后，未发现缺陷。上传音乐功能和创建专辑功能正常。

## 1.3.3链接测试

链接是web应用系统的一个主要特征，它是在页面之间切换和引导用户去一些未知地址页面的主要手段。

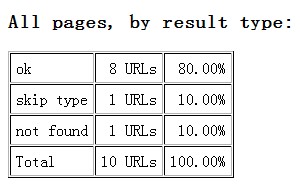
* 1. 链接测试原理

链接测试的原理是：从待测网站的根目录开始搜索所有的网页文件，对所有网页文件中的超级链接、图片文件、包含文件、CSS文件、页面内部链接等所有连接进行读取。如果网站文件不存在、指定文件链接不存在或者是指定页面不存在，则将该链接和所处的集体位置记录下来。如果发现被测试网站内有页面既没有连接到其他资源也没有被其他资源链接，则可以判定该页面为孤立页面，将该页面添加到孤立页面记录中。

* 1. 音乐分享站的链接测试

使用Xenu Link Sleuth，进入其主界面，输入网站地址进行测试。测试完毕后，可以过Report得到测试报告。

Xenu执行完毕测试后，会自动生成测试报告，测试报告中会列出个链接的URL，并分析网站中的链接情况，给出测试结果。本次测试的结果如下图所示：



# 1.4 校内音乐分享站系统性能测试

性能测试就是模拟大量用户对软件系统的各种操作，获取系统和应用的性能指标，分析软件是否满足用户的需求。性能需求的特点决定了我们不可能完全采用传统的手工方式完成，必须借助于自动化测试工具来实现。自动化测试工具具有以下特点和优势：

（1）通过使用虚拟用户代替实际用户来减少测试人员数量上的要求。虚拟用户模拟实际用户行为，运行实际的应用程序。

（2）一台计算机上可以运行多个虚拟用户，因此减少了硬件设备的需求。

（3）可以是测试人员从一个单一的控制点简单有效地控制所有虚拟用户。

（4）在测试过程中会自动记录应用程序的性能。测试人员可以从众多图表和报告中进行选择以查看性能数据。

本系统采用HP-Mercury公司的LoadRunner性能测试工具进行性能测试。LoadRunner的测试流程如下：

第一步

制定测试计划

创建虚拟用户脚本

创建场景

运行测试

监视场景

分析测试结果

第二步

第三步

第四步

第五步

第六步

1. **制定测试计划**

在任何类型的测试中，测试计划都是必要的步骤。测试计划是进行成功的负载测试的关键。在任何类型的测试的第一步都是制定比较详细的测试计划。

测试计划中要重点分析系统的性能需求，并确定性能测试内容。

1. **开发测试脚本**

LoadRunner使用虚拟用户的活动模拟真实用户来操作Web应用程序，而虚拟用户的活动就包含在测试脚本中，所以说测试脚本对于测试来说是非常重要的。

开发测试脚本要使用VuGen组件。测试脚本要完成的内容：

（1）创建每一个虚拟用户的活动。

（2）定义事务。

（3）设定集合点。

（4）插入检查点。

1. **创建场景**

场景描述在测试活动中发生的各种事件。一个场景包括一个运行虚拟用户活动的压力机器（Load Generator）列表，一个测试脚本的列表以及大量的虚拟用户和虚拟用户组。我们使用LoadRunner中的Controller来创建场景。在创建测试场景之前需要设计出合理的测试场景，以便更真实地模拟用户使用软件系统的情况。

1. **运行测试**

一切配置妥当即可开始运行测试。

1. **监视场景**

在运行测试的过程中，需要监视各个服务器的运行情况（数据库服务器、Web服务器等）和客户端的情况，获取各项性能指标。件事各项性能指标可通过添加性能计数器来实现。

1. **分析测试结果**

我们需要分析大量的图表，生成各种不同的报告，最后得出结论，并对系统性能作出评价。

下面按照性能测试的流程详细介绍校内音乐分享站的性能测试过程。

## 1.4.1测试用例设计

**登录模块测试用例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | 登录校内音乐分享站测试用例 | | | | |
| 功能 | 系统支持多个用户并发登录 | | | | |
| 目的 | 测试多用户登录时系统的处理能力 | | | | |
| 方法 | 模拟多个用户在不同客户端登录，然后并发进入系统。采用LoadRunner录制登录过程（ | | | | |
| 并发用户数与事务执行情况 | | | | | |
| 并发用户数 | 事务平均响应时间 | 事务最大响应时间 | 事务成功率 | 每秒点击率 | 平均流量（字节/秒） |
| 15 | 81.476 | 141.658 | 0.144 | 5.254 | 5.013 |
| 10 | 72.675 | 163.380 | 0.05 | 3.874 | 3.931 |
| 5 | 34.186 | 49.873 | 0.130 | 4.672 | 4.483 |

**上传音乐测试用例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | 上传音乐测试用例 | | | | |
| 功能 | 系统支持多个用户并发上传音乐 | | | | |
| 目的 | 测试多个用户同时上传音乐时系统的处理能力 | | | | |
| 方法 | 模拟多个用户在不同客户端上传音乐，然后并发操作。采用LoadRunner录制上传音乐过程，然后利用其完成测试，要监视服务器的性能。 | | | | |
| 并发用户数与事务执行情况 | | | | | |
| 并发用户数 | 事务平均响应时间 | 事务最大响应时间 | 事务成功率 | 每秒点击率 | 平均流量 |
| 10 | 258．7 | 389．6 | 96% | 3．463 | 3.110 |
| 20 | 290．8 | 408．3 | 97% | 2.325 | 4.421 |

**创建专辑测试用例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | 创建专辑测试用例 | | | | |
| 功能 | 系统支持多个用户并发创建专辑 | | | | |
| 目的 | 测试多个用户同时创建专辑时系统的处理能力 | | | | |
| 方法 | 模拟多个用户在不同客户端创建专辑，然后并发操作。采用LoadRunner录制创建专辑过程，然后利用其完成测试，要监视服务器的性能。 | | | | |
| 并发用户数与事务执行情况 | | | | | |
| 并发用户数 | 事务平均响应时间 | 事务最大响应时间 | 事务成功率 | 每秒点击率 | 平均流量 |
| 10 | 212．1 | 325．4 | 0.013 | 4.324 | 4.232 |
| 20 | 243．7 | 454．3 | 0.108 | 2.433 | 2.322 |

## 1.4.2测试环境

**性能测试软/硬件配置**

|  |  |  |
| --- | --- | --- |
| 设备 | 硬件配置 | 软件配置 |
| 服务器 |  |  |
| 负载产生设备（模拟客户端） |  |  |

## 1.4.3测试执行

**执行测试场景**

## 1.4.4测试结果分析

由测试结果数据可知，随着并发事件数量的增加，响应时间明显越来越慢，超过一定数量时，系统将无法接受，但是由于试验了条件有限，暂时没有测出其限制具体数量。