上海双地信息系统有限公司

2019/10/31

上海图书馆  
全文OCR等检索功能项目

性能测试报告

目 录

[1. 测试工具 3](#_Toc23973751)

[2. 测试对象 3](#_Toc23973752)

[3. 测试环境 4](#_Toc23973753)

[4. OCR检索 4](#_Toc23973754)

[4.1. 测试场景 4](#_Toc23973755)

[4.1.1. 场景1 : 检索 4](#_Toc23973756)

[4.2. 测试结果 4](#_Toc23973757)

[4.2.1. 结果1 : OCR检索 4](#_Toc23973758)

[5. 整本浏览服务 5](#_Toc23973759)

[5.1. 测试场景 5](#_Toc23973760)

[5.1.1. 场景1 : 数据转换效率 5](#_Toc23973761)

[5.2. 测试结果 6](#_Toc23973762)

[5.2.1. 结果1 : 数据转换效率 6](#_Toc23973763)

[6. 镜像站 6](#_Toc23973764)

[6.1. 测试场景 6](#_Toc23973765)

[6.1.1. 场景1 : 检索压力 6](#_Toc23973766)

[6.1.2. 场景2 : 数据更新性能 (非loadrunner执行) 6](#_Toc23973767)

[6.1.3. 场景3 : 服务代理性能验证 6](#_Toc23973768)

[6.2. 测试结果 7](#_Toc23973769)

[6.2.1. 结果1 : 检索 7](#_Toc23973770)

[6.2.2. 结果2 : 更新 8](#_Toc23973771)

[6.2.3. 结果3 : 服务代理性能验证 8](#_Toc23973772)

# 测试工具

LoadRunner 是一种预测系统行为和性能的工业标准级负载测试工具。通过以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。通过使用 LoadRunner ， 企业能最大限度地缩短测试时间， 优化性能和加速应用系统的发布周期。目前企业的网络应用环境都必须支持大量用户，网络体系架构中含各类应用环境且由不同供应商提供软件和硬件产品。难以预知的用户负载和愈来愈复杂的应用环境使公司时时担心会发生用户响应速度过慢， 系统崩溃等问题。这些都不可避免地导致公司收益的损失。Mercury Interactive 的 LoadRunner 能让企业保护自己的收入来源，无需购置额外硬件而最大限度地利用现有的 IT 资源， 并确保终端用户在应用系统的各个环节中对其测试应用的质量， 可靠性和可扩展性都有良好的评价。LoadRunner 是一种适用于各种体系架构的自动负载测试工具， 它能预测系统行为并优化系统性能。LoadRunner 的测试对象是整个企业的系统， 它通过模拟实际用户的操作行为和实行实时性能监测， 来帮助您更快的查找和发现问题。此外，LoadRunner 能支持广范的协议和技术， 为您的特殊环境提供特殊的解决方案。

# 测试对象

1. OCR检索
2. 整本浏览
3. 镜像站服务平台

# 测试环境

|  |  |  |  |
| --- | --- | --- | --- |
| 机器型号 | 配置信息 | 数量 | 备注 |
| PC | * 2.2GHz \*2； * 2GB内存\*2； * 500G * 网卡：100M | 1台 | 服务器 |

# OCR检索

## 测试场景

### 场景1 : 检索

检索任务，存在多人同时使用检索的情况，存在高并发可能，需要测试系统性能。

1. 模拟用户登陆；
2. 使用不同关键字对OCR信息进行检索。

## 测试结果

### 结果1 : OCR检索

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | | OCR检索 | | | | | |
| 目的 | | 测试多个用户同时对系统并发操作时系统的性能。 | | | | | |
| 方法 | | * 1. 模拟用户名密码登陆   2. 模拟点击数据检索 | | | | | |
| 并发用户数与事务执行情况 | | | | | | | |
| 并发用户数 | | | 20 | 50 | 100 | 300 | 500 |
| 普通检索 | 通过的虚拟用户数 | | 20 | 50 | 100 | 300 | 500 |
| 事务平均响应时间（s） | | 0.71 | 3.8 | 3.47 | 5.15 | 6.70 |
| 事务最大响应时间（s） | | 1.0 | 5.2 | 5.75 | 10.1 | 11.6 |
| 平均每秒通过事务数 | | 3.15 | 14.81 | 16.08 | 24.2 | 25.9 |
| 每秒点击次数 | | 53.720 | 251.81 | 273.39 | 412.62 | 440.50 |
| 每秒HTTP响应数 | | 53.720 | 251.81 | 273.39 | 412.62 | 440.50 |

# 整本浏览服务

## 测试场景

### 场景1 : 数据转换效率

1. SQL数据入ORACLE，300万数据量测试
2. ORACLE数据如SOLR，300玩数据量测试

## 测试结果

### 结果1 : 数据转换效率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 平均（s） |
| SQL=>ORACLE | 3776 | 3716 | 3833.6 | 3981.6 | 3655.2 | 3792.4 |
| ORACLE=>SOLR | 2683.5 | 2601.2 | 2787.1 | 2643.2 | 2558.64 | 2654.7 |

# 镜像站

## 测试场景

### 场景1 : 检索压力

检索服务，因为普通检索、高级检索、专业检索的核心检索部分是一致的，我们选择容易跑压力测试的普通检索服务作为代表；

* 用户采用IP自动登陆方式，免除了登陆步骤
* 使用检索路径直接进行检索，等待结果页面的返回。

### 场景2 : 数据更新性能 (非loadrunner执行)

数据更新是镜像站更新数据的性能测试，

* 为测试镜像站提供更新包，数据包含80万索引
* 在镜像站进行数据更新，统计执行时间。

### 场景3 : 服务代理性能验证

* 采用Lighttpd作为代理服务
* 采用Nginx做为代理服务

验证静态资源（图片）和动态资源（检索页）访问的效率差异，

## 测试结果

### 结果1 : 检索

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | | 核心的检索性能 | | | | | |
| 目的 | | 测试多个用户同时对系统并发操作时系统的性能。 | | | | | |
| 方法 | | * 1. 直接访问普通检索地址，采用IP登陆认证   2. 点击聚类进行进一步检索   3. 点击详情页进行查看 | | | | | |
| 并发用户数与事务执行情况 | | | | | | | |
| 并发用户数 | | | 20 | 50 | 100 | 300 | 500 |
| 普通检索 | 通过的虚拟用户数 | | 20 | 50 | 100 | 300 | 500 |
| 事务平均响应时间（s） | | 0.302 | 1.63 | 1.466 | 2.175 | 2.828 |
| 事务最大响应时间（s） | | 0.452 | 2.23 | 2.429 | 4.282 | 4.924 |
| 平均每秒通过事务数 | | 1.333 | 6.25 | 6.786 | 10.241 | 10.933 |
| 每秒点击次数 | | 22.667 | 106.25 | 115.357 | 174.103 | 185.867 |
| 每秒HTTP响应数 | | 22.667 | 106.25 | 115.357 | 174.103 | 185.867 |

### 结果2 : 更新

对80万数据的3次更新结果如下所示

|  |  |
| --- | --- |
| 次数 | 耗时（秒） |
| 第一次 | 652s |
| 第二次 | 623s |
| 第三次 | 638s |
| 平均耗时 | 637s |

### 结果3 : 服务代理性能验证

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | | 静态资源访问 | | | | | |
| 目的 | | 测试多个用户同时对系统并发操作时系统的性能。 | | | | | |
| 方法 | | * 1. 直接访问公共图片资源路径   2. 直接访问公共js静态脚本 | | | | | |
| 并发用户数与事务执行情况 | | | | | | | |
| 并发用户数 | | | 20 | 50 | 100 | 300 | 500 |
| 静态资源访问(lighttpd) | 通过的虚拟用户数 | | 20 | 50 | 100 | 300 | 500 |
| 事务平均响应时间（s） | | 0.097 | 0.59 | 0.64 | 0.77 | 0.92 |
| 事务最大响应时间（s） | | 0.15 | 0.75 | 0.84 | 1.40 | 1.86 |
| 平均每秒通过事务数 | | 2.67 | 14.7 | 16.4 | 23.0 | 25.9 |
| 每秒点击次数 | | 54.1 | 253.8 | 263.4 | 392.5 | 437.7 |
| 每秒HTTP响应数 | | 54.1 | 253.8 | 263.4 | 392.5 | 437.7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | | 静态资源访问 | | | | | |
| 目的 | | 测试多个用户同时对系统并发操作时系统的性能。 | | | | | |
| 方法 | | * 1. 直接访问公共图片资源路径   2. 直接访问公共js静态脚本 | | | | | |
| 并发用户数与事务执行情况 | | | | | | | |
| 并发用户数 | | | 20 | 50 | 100 | 300 | 500 |
| 静态资源访问(nginx) | 通过的虚拟用户数 | | 20 | 50 | 100 | 300 | 500 |
| 事务平均响应时间（s） | | 0.10 | 0.52 | 0.57 | 0.86 | 1.08 |
| 事务最大响应时间（s） | | 0.16 | 0.71 | 0.84 | 1.46 | 1.85 |
| 平均每秒通过事务数 | | 2.66 | 13.2 | 13.9 | 20.4 | 26.3 |
| 每秒点击次数 | | 52.8 | 218.6 | 242.1 | 382.1 | 419.9 |
| 每秒HTTP响应数 | | 52.8 | 218.6 | 242.1 | 382.1 | 419.9 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | | 动态资源访问 | | | | | |
| 目的 | | 测试多个用户同时对系统并发操作时系统的性能。 | | | | | |
| 方法 | | * 1. 按照检索脚本进行执行 | | | | | |
| 并发用户数与事务执行情况 | | | | | | | |
| 并发用户数 | | | 20 | 50 | 100 | 300 | 500 |
| 动态资源访问(lighttpd) | 通过的虚拟用户数 | | 20 | 50 | 100 | 300 | 500 |
| 事务平均响应时间（s） | | 0.30 | 1.56 | 1.25 | 1.94 | 2.59 |
| 事务最大响应时间（s） | | 0.44 | 2.06 | 2.27 | 4.10 | 4.32 |
| 平均每秒通过事务数 | | 1.16 | 5.28 | 6.35 | 9.56 | 8.99 |
| 每秒点击次数 | | 18.47 | 85.08 | 106.08 | 159.16 | 162.73 |
| 每秒HTTP响应数 | | 18.47 | 85.08 | 106.08 | 159.16 | 162.73 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | | 动态资源访问 | | | | | |
| 目的 | | 测试多个用户同时对系统并发操作时系统的性能。 | | | | | |
| 方法 | | * 1. 按照检索脚本进行执行 | | | | | |
| 并发用户数与事务执行情况 | | | | | | | |
| 并发用户数 | | | 20 | 50 | 100 | 300 | 500 |
| 动态资源访问(nginx) | 通过的虚拟用户数 | | 20 | 50 | 100 | 300 | 500 |
| 事务平均响应时间（s） | | 0.25 | 1.41 | 1.22 | 2.15 | 2.45 |
| 事务最大响应时间（s） | | 0.39 | 1.87 | 2.37 | 3.62 | 4.65 |
| 平均每秒通过事务数 | | 1.12 | 6.03 | 6.04 | 9.54 | 9.63 |
| 每秒点击次数 | | 22.52 | 91.84 | 104.12 | 147.47 | 157.48 |
| 每秒HTTP响应数 | | 22.52 | 91.84 | 104.12 | 147.47 | 157.48 |

结论：在我们应用中，lighttpd和nginx的性能差不多，平均性能相差在5%以内浮动。