# 图片服务器下载性能测试报告

**1 概述**

**1.1 目的**

本测试报告为高手后台的性能测试报告，目的在于总结测试阶段的测试以及分析测试结果，描述网站是否符合需求。

**1.2 背景**

高手，本公司目前正在进行性能测试。考虑到用户数量及数据的增多给服务器造成压力不可估计，因此计划对高手app后台负载性能测试，在系统配置不变的情况下，在一定时间内，服务器在高负载情况下的性能行为表现，便于对系统环境进行正确的分析及评估。

**1.3 范围**

本次测试主要是高手app后台接口系统的性能测试。

**1.1 引用文档**

下表列出了执行测试过程所引用的文档：

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称 | 版本号 | 作者 | 备注 |
| 后台junittest java程序 |  | 高手后台开发人员 |  |
| 性能测试总结 |  | 冯炳发 |  |
|  |  |  |  |
|  |  |  |  |

**2 测试概要**

**2.1 测试环境**

下图描述测试该项目所需要的硬件环境：

|  |  |
| --- | --- |
| 客户机 | Intel(R)  TM) CUP  四核i3处理器、内存：4GB RAM |
| 服务器 | I3 4G内存 4核cpu 硬盘常规硬盘，由于实际情况限制，将业务服务器和数据库服务部署在同一机器上，全部没有优化，使用系统软件默认配置。 |
|  |  |

下图描述测试网络的拓扑结构：



      客户机测试环境                                 服务器测试环境

测试机与被测服务器在同一局域网通过网线直接连接进行，排除了网速度不稳定性。

系统采用B/S架构模式，客户端通过中间件访问数据库，中间件和数据库现在部署在一台服务器上。

网速通过在服务器端安装wondershaper 对应版本，限制网速想、上行下行均为 5000kb 即5M

**2.2 人力资源**

下表列出了所有参与此项目的测试人员：

|  |  |
| --- | --- |
| 角色 | 资源数量/具体人员 |
| 测试员 | 肖雨龙 |

**2.1 测试工作量**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 任务 | 开始时间 | 结束时间 | 总计（天数） | 总计（人时） |
| 计划 | 2013-11-19 | 2011-11-29 | 10 |  |
| 实际 | 2013-11-22 | 2011-12-1 | 9 |  |
|  |  |  |  |  |

**3 测试内容及方法**

**3.1 测试需求/目标**

在大用户量、数据量的超负荷下，获得服务器运行时的相关数据，从而进行分析，找出系统瓶颈，为后续优化提供参考

**3.2 测试内容**

本次测试主要是对高手app后台接口的访问操作在大负荷情况下处理数据的能力及承受能力。 均不考虑图片服务器，因为图片服务器是单独占据一个服务器以及相应带宽。具体数据详见缩略结果excel。

具体的接口分析，可以实际沟通（由于实际关注的只是性能不佳的，其他的可以实际沟通）

测试方法：

注释：所有用户登陆、没有权限限制。

**3.3 测试工具**

主要测试工具为：LoadRunner性能测试工具11

辅助软件：截图工具，Word，监视系统工具脚本（修改添加了具体的网速统计）

4 测试结果及分析

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 场景 | 系统压力 | 成功率 | 响应时间 | TPS | cpu空闲率 | | 内存空闲率  proxy 6G，app 4G | |
|  | url | 平均值 | 平均值 | 平均值 | Proxy | APP | Porxy | APP |
| 50并发 | 150并发（每个user实际为三张图片） | 99.89% | 0.396 | 95.66 | 89% | 76% | 87.3% | 61.75% |

 从测试结果来看，可以得到如下结论：

1、随着并发数的增加，系统响应时间增加，但是都在0.5秒以内

2、随着并发数的增加，proxy，app的内存，CPU空闲率也会下降，且proxy的下降幅度明显小于APP，当达到3000并发，app压力达到1500时，app 的cpu下降到74%左右，接近安全临界值 70%， 当app上的并发数达到2000时，cpu空闲率降为67%，无法满足需求，且内存波动幅度增大，说明proxy+app系统性能瓶颈在APP

随着并发数的增加，TPS会增加，但是增加到95左右后不再上升，说明系统的最大TPS为95

4、相同并发下每秒上线用户数不同，测试结果大致相同，说明每秒上线数 对系统性能没有影响（如果每秒上线数达到了系统能够承受的最大值，此时会有影响）

**4.1 处理性能评估**

这次测试属于局域网环境进行，排除了外网的不稳定性。

**并发用户测试**

测试内容：

这次测试属于模拟真实环境；

说明：用户的整个执行流程都录制在Action(循环)部分，所以Vuser\_int (开始)和Vuser\_end(结束)部分为空。Action\_Transaction部分的时间为运行整个Action脚本所需的时间。

整个Action的平均响应时间详细参见测试场景。

# 5测试结果及分析

5.1 本次测试，平均响应时间不满足要求。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **场景** | **系统压力** | | | **成功率** | | **响应时间** | | **TPS** | | **cpu空闲率** | | | **内存空闲率**  **proxy 6G，app 4G** | | |
|  | proxy | app\_1 | app\_2 | | 平均值 | | 平均值 | | 平均值 | | Proxy | APP | | Porxy | APP |
| 50并发 | 3000并发 | 1500并发 | 1500并发 | | 99.89% | | 0.396 | | 95.66 | | 89% | 76% | | 87.3% | 61.75% |

总体测试数据统计如下

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **场景** | **系统压力** | | | **成功率** | | **响应时间** | | **TPS** | | **cpu空闲率** | | | **内存空闲率**  **proxy 6G，app 4G** | |
|  | proxy | app\_1 | app\_2 | | 平均值 | | 平均值 | | 平均值 | | Porxy | APP | Porxy | APP |
| 1000并发，每秒上线200个，运行1小时 | 1000并发 | 500并发 | 500并发 | | 100% | | 0.057 | | 84.92 | | 95% | 90% | 85.00% | 66.00% |
| 2000并发，每秒上线200个，运行1小时 | 2000并发 | 1000并发 | 1000并发 | | 99.99% | | 0.154 | | 95.57 | | 92% | 82% | 86.30% | 69.50% |

从测试结果来看，可以得到如下结论：

1、随着并发数的增加，系统响应时间增加，但是都在0.5秒以内

2、随着并发数的增加，proxy，app的内存，CPU空闲率也会下降，且proxy的下降幅度明显小于APP，当达到3000并发，app压力达到1500时，app 的cpu下降到74%左右，接近安全临界值 70%， 当app上的并发数达到2000时，cpu空闲率降为67%，无法满足需求，且内存波动幅度增大，说明proxy+app系统性能瓶颈在APP

随着并发数的增加，TPS会增加，但是增加到95左右后不再上升，说明系统的最大TPS为95

4、相同并发下每秒上线用户数不同，测试结果大致相同，说明每秒上线数 对系统性能没有影响（如果每秒上线数达到了系统能够承受的最大值，此时会有影响）

说明：Java 垃圾回收机制可能在压力执行时回收内存，这样导致内存占用情况无法体现出并发用户增加时，对内存的趋势，测试中主要观察内存增加的幅度和是否持续增加导致内存溢出。

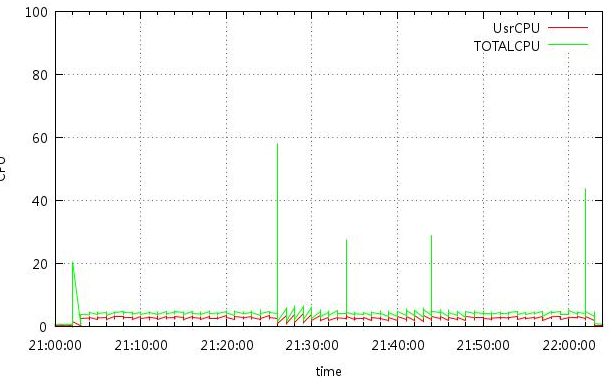
## 5.2测试数据统计和资源分析

### *1000并发，每秒上线200个，运行1小时*

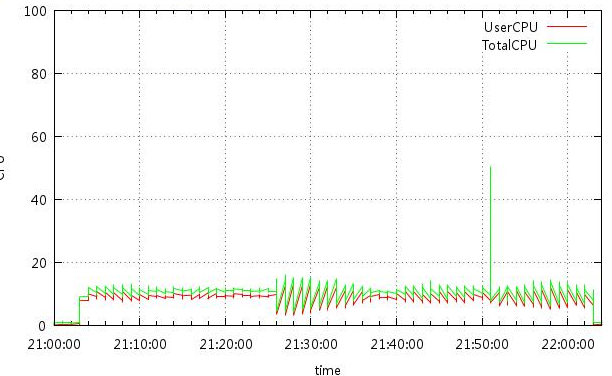
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **场景** | **系统压力** | | | **成功率** | | **响应时间** | | **TPS** | | **cpu空闲率** | | | **内存占用（G）** | |
|  | proxy | app\_1 | app\_2 | | 平均值 | | 平均值 | | 平均值 | | Porxy | APP | Porxy | APP |
| 1000并发，每秒上线200个，运行1小时 | 1000并发 | 500并发 | 500并发 | | 100% | | 0.057 | | 84.92 | | 95% | 90% | 85.00% | 66.00% |

1000用户并发时，成功率为100%，平均响应时间为0.057，服务器等资源也保持在良好的状态。

Proxy CPU资源信息

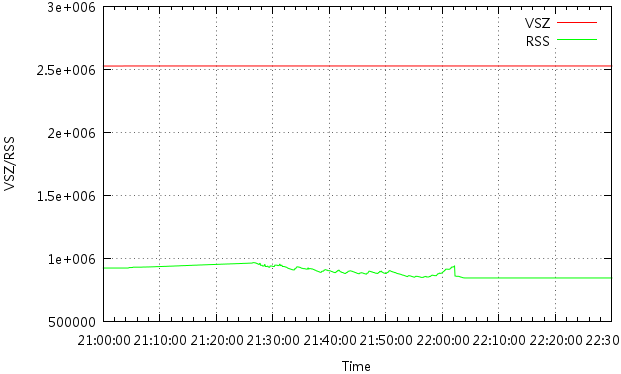


APP CPU资源信息图

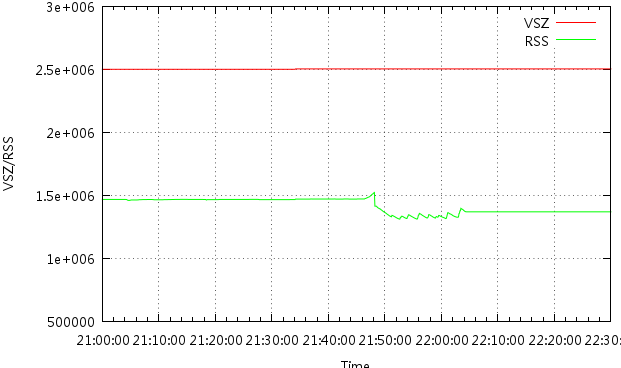


从以上CPU资源图可以看出，Proxy性能消耗基本保持在5%左右，APP资源消耗保持在10%左右，在压力测试过程虽然出现过瞬间峰值的情况，但峰值没有持续，满足性能要求。

Proxy 程序占用的虚拟内存（VSZ）和实际内存（RSS）大小



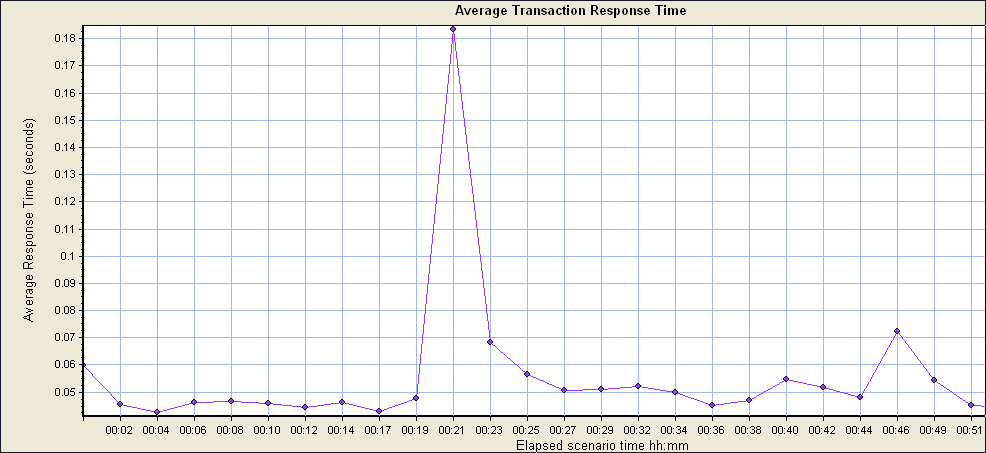
APP 程序暂用的虚拟内存（VSZ）和实际内存（RSS）大小



Proxy和APP的 VSZ稳定在2.5G，说明程序分配的物理内存已经能够满足程序的运行，且RSS前期保持在1.5G左右，后期还存在下降的趋势。

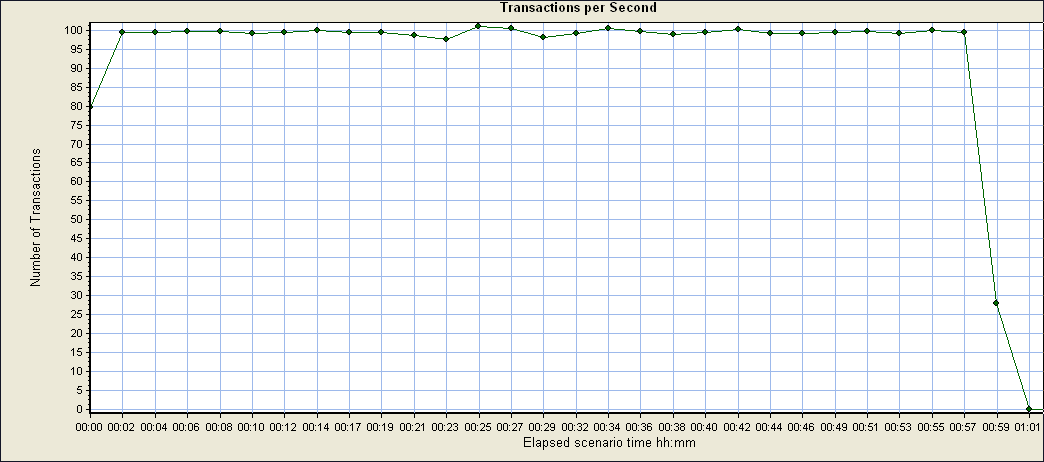
说明：经验数据显示对于4G内存的系统，当RSS达到3G后系统会变得不稳定

平均响应时间



平均响应时间出现一次响应时间峰值，与CPU，内存出现的峰值情况基本对应，在CPU，内存出现峰值时，系统的处理速度下降，响应时间也随之上升。

每秒通过的事物数



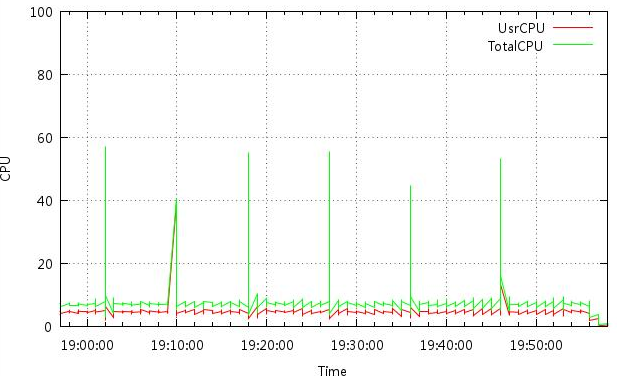
每秒通过的事物数非常稳定，且保持在100左右，因此并发1000时系统的平均处理请求能力为100左右。

### *2000并发，每秒上线200个，运行1小时*

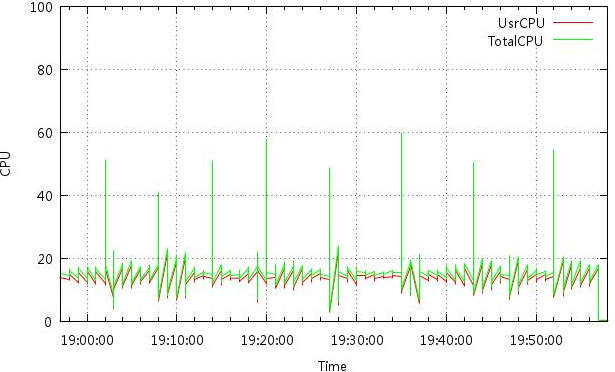
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **场景** | **系统压力** | | | **成功率** | | **响应时间** | | **TPS** | | **cpu空闲率** | | | **内存占用（G）** | |
|  | proxy | app\_1 | app\_2 | | 平均值 | | 平均值 | | 平均值 | | Porxy | APP | Porxy | APP |
| 2000并发，每秒上线200个，运行1小时 | 2000并发 | 1000并发 | 1000并发 | | 99.99% | | 0.154 | | 95.57 | | 92% | 82% | 86.30% | 69.50% |

2000个用户并发，每秒上线200个时，各种性能数据也满足性能要求，较1000个并发的场景在CPU和内存消耗方面有所增加

Proxy CPU资源信息

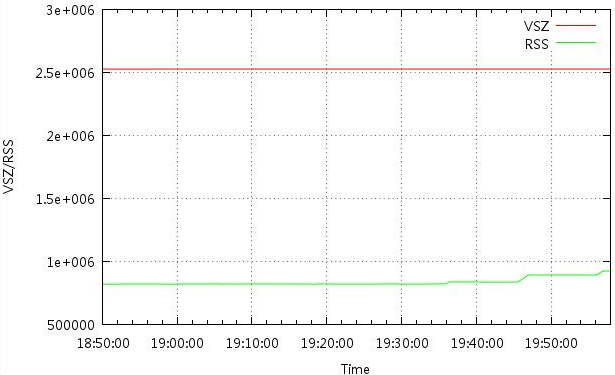


APP CPU信息

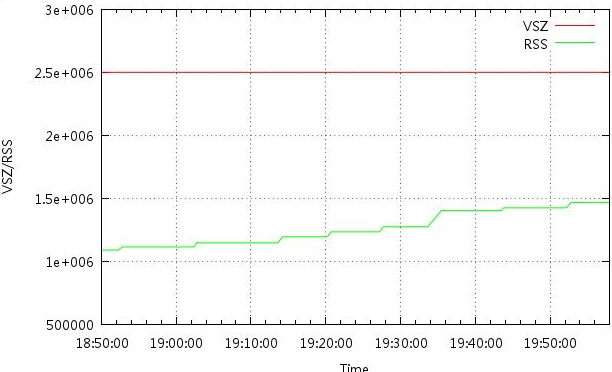


2000个用户并发场景下，CPU消耗增加，同时出现峰值的情况也增多，但总体来说比较稳定，满足性能要求。

Proxy 程序占用的虚拟内存（VSZ）和实际内存（RSS）大小

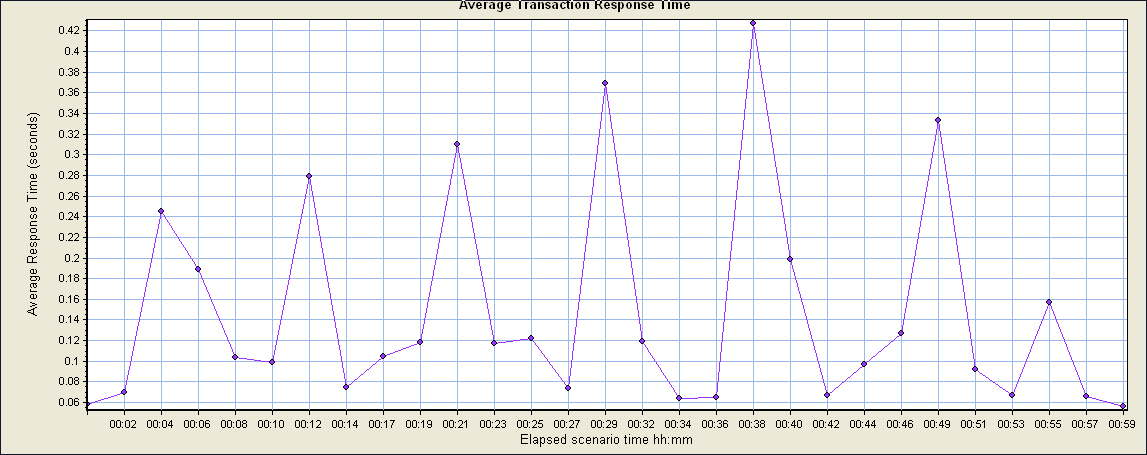


APP 程序占用的虚拟内存（VSZ）和实际内存（RSS）大小



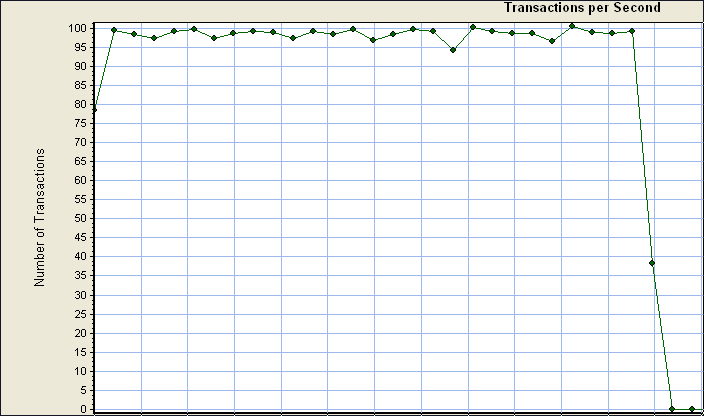
2000并发时（实际APP承受的是1000并发），Proxy 和APP的内存都在增加，APP 增幅明显大于Proxy，Proxy增加了200M左右，APP增加了800M左右，但总内存消耗在1.5G以下

平均事务响应时间



与1000并发情况相同，CPU，内存出现峰值后，平均响应时间对应出现了峰值，响应时间波动较大。

每秒点击率



## 附件

## Addstep，updateIcons：

实际的场景描述： 假设会场有5k人会在大约5min之内访问后台，大约考虑到8020定律，即一分钟之内会有50人左右会访问后台，而考虑到实际使用app时大部分人的操作是查看app，而不是创建新的课程，现在可以同时承受10个人处理，在8s内处理完成，即每天估计约有5小时会有人满载上传图片的情况，上传接口的容量约为22500/每天，单接口的情况符合当前的用户容量。

Login/login(包含注册)：

单接口50人并发访问情况，响应时间超过10s，考虑到实际的情况。4K人约在一分钟左右注册，情况会导致响应时间较慢。

Workcomment：

作品评论，

此接口30人的压力下进行访问，单接口平均12s，但是注意时间一直递增到24s，考虑到实际的使用情况，此接口的使用应该在降低失败业务水平之后将此接口的承载能力增加到约50，

Sendpmessage

此接口为私信接口，使用人数预估计压力应该约为30人以上的并发，当前的实际承载能力是小于此值的，考虑到实际的硬件占用，带宽占用较低的情况， 需要优化这部分。

Updateuser：

此接口主要用于更新信息，使用情况较少，实际能够承受约4K人6分钟内访问人数的平均压力约为 10人/s左右。

图片访问：

实际的测试时在50人并发访问下进行的，测试结束后查看日志，发现由于测试准备的图片较少，实际的响应时间应该是优于实际的服务器的，此处的响应时间平均约为0.7s但是考虑到实际的app 图片页面访问情况，实际应该承受 打开主页数\*app数量 5k8分钟的情况下，大约峰值的压力约为800次/S，因而对于图片服务器的IO压力，带宽压力预估计都是超过当前的承载能力的，这部分需要优化Apache2配置，提升带宽。由于还没有测试阿里云机器的标准io和cpu内存性能，此处的硬件优化需要接下来实际进行讨论验证。

# 6附录

### 5.1主要的测试脚本：



#### 5.2测试的文档：



