**单点登录服务单体测试报告书**

文档管理信息表

|  |  |
| --- | --- |
| **主题** | 单点登录服务单体测试报告书 |
| **版本** | 1.0 |
| **内容** |  |
| **关键字** |  |
| **参考文档** |  |
| **创建时间** | 2019-11-14 |
| **创建人** | 王雷 |
| **最新发布日期** |  |

文档变更记录表

|  |  |  |
| --- | --- | --- |
| **修改人** | **修改时间** | **修改内容** |
| 王雷 | 2019-11-20 | 初版完成 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**目 录**

[1 测试说明 1](#_Toc25752555)

[1.1 测试项说明 1](#_Toc25752556)

[1.2 测试角色简介 1](#_Toc25752557)

[1.3 测试角色关系图 2](#_Toc25752558)

[1.4 测试环境配置 2](#_Toc25752559)

[**1.4.1** Login Server运行环境： 2](#_Toc25752560)

[**1.4.2** 虚拟角色运行环境： 3](#_Toc25752561)

[**1.4.3** 网络连接 3](#_Toc25752562)

[1.5 测试模式 3](#_Toc25752563)

[**1.5.1** 接口数据准确性测试模式 3](#_Toc25752564)

[**1.5.2** 性能测试模式 3](#_Toc25752565)

[2 接口数据准确性联合测试 3](#_Toc25752566)

[2.1 测试方法 3](#_Toc25752567)

[2.2 测试结果截图 4](#_Toc25752568)

[**2.2.1** 模拟数据源执行结果 4](#_Toc25752569)

[**2.2.2** 模拟QB客户端执行结果 5](#_Toc25752570)

[**2.2.3** 模拟网关服务执行结果 5](#_Toc25752571)

[**2.2.4** 模拟DDS服务执行结果 6](#_Toc25752572)

[2.3 测试结论 6](#_Toc25752573)

[3 服务性能测试 6](#_Toc25752574)

[3.1 lava box通信方式的性能测试 6](#_Toc25752575)

[**3.1.1** 模拟QB客户端的性能测试 6](#_Toc25752576)

[**3.1.2** 模拟QB客户端的性能测试 10](#_Toc25752577)

[**3.1.3** 基于lava box业务性能测试结论 12](#_Toc25752578)

[3.2 msg bus通信方式的性能测试 13](#_Toc25752579)

[**3.2.1** 模拟网关服务的性能测试 13](#_Toc25752580)

[**3.2.2** 基于msgbus业务性能测试结论 14](#_Toc25752581)

[4 单例服务运行稳定性测试 14](#_Toc25752582)

[5 多实例服务数据同步验证 15](#_Toc25752583)

[6 服务高可用测试 15](#_Toc25752584)

# 测试说明

## 测试项说明

单点登录服务的测试点主要集中在对此服务的高可用性、性能、数据准确性等方面，按测试步骤的先后顺序，测试项主要为如下几点：

1. 单实例服务各个接口数据准确性验证。
2. 单实例服务性能测试。
3. 单实例服务运行稳定性测试。
4. 高可用架构服务各实例之间的数据同步验证。
5. 服务的高可用测试。

## 测试角色简介

单点登录服务的数据交互对象很多，有QB、QM、Excel等客户端，有后台网关服务gateway server，还有DDS server，QB web组件，用户账号信息数据源服务等，因此为配合测试这个服务，需要模拟出这些测试角色。

目前已编码实现的测试角色如下：

1. user\_info\_supplier\_mock，该角色模拟用户账号信息的数据源服务，可以通过设定创建出指定数目的用户账号数据，然后通过msgbus的广播接口发送给单点登录服务(login server)，用于用户身份验证。
2. qb\_clients\_mock，该角色模拟QB、QM、Excel等客户端，这些客户端向login server发起登录请求，服务将验证其身份后返回相应的结果。
3. gtw\_server\_mock，该角色模拟后台网关服务gateway server，它将收到qb\_clients\_mock发来的用户账号信息和token，并使用这些信息向login server发起token验证，根据验证结果决定是否允许QB client端接入后台服务。
4. dds\_server\_mock，该角色模拟DDS网关服务DDS server，它也将收到qb\_clients\_mock发来的用户账号信息和token，并使用这些信息向login server发起token验证，根据验证结果决定是否允许QB client端接入DDS网络。

## 测试角色关系图



1．测试对象：Login Server运行于CentOS7虚拟机，其他所有模拟对象都是为了测试这个服务设计的。

2．数据源模拟：user\_info\_supplier\_mock运行在Win7，创建指定数量的用户账户信息，然后通过msgbus的fanout广播接口发送给Login Server，后者将这些用户数据缓存，之后会用于客户端登录验证、token验证等场合，此角色执行完可以退出。

3．QB客户端模拟：qb\_clients\_mock运行在Win7，它模拟QB客户端向Login Server发起登录验证请求，后者将验证其身份合法后会返回token等信息，然后qb\_clients\_mock会通过share memroy的方式，将账户id和token发送给dds\_server\_mock和gtw\_server\_mock。

4．网关服务模拟：gtw\_server\_mock与qb\_clients\_mock运行在同一台机器上，它将通过share memory接收用户id和token信息，然后使用这些信息向Login Server发起token验证请求，后者会返回验证结果。

5．DDS服务模拟：dds\_server\_mock与qb\_clients\_mock运行在同一台机器上，它也通过share memory接收客户端发过来的用户id和token信息，然后用其向Login Server发起token验证请求，后者会返回验证结果。

6．注意：在实际场景中，QB客户端会通过msgbus与gateway server通信发送账号和token，也会通过lava box与dds server通信发送账号和token，本测试将这两种通信方式使用share memory的方式代替是为了测试更加便捷，且因这两种通信过程与测试Login Server并无直接关联。

## 测试环境配置

### Login Server运行环境：

CentOS7.7虚拟机：

CPU：Intel Core i5-4460 3.2G 物理双核 4个超线程

内存：4G

### 虚拟角色运行环境：

Windows10笔记本：

CPU：Intel Core i5-8350U 1.7G 物理4核 8个超线程

内存：8G

### 网络连接

以太网，最大带宽1Gbps

## 测试模式

测试模式分为两种，一种是为验证各接口数据准确性而设计，另外一种是为测试login server性能而设计。

第一种模式会放慢速度，可以逐一验证各接口的输入输出参数是否准确。

第二种模式不解码消息，收到消息后仅仅记录统计数字，强调测试服务并发处理大量数据的性能。

### 接口数据准确性测试模式

该模式下qb\_clients\_mock速度放慢，每秒发起一次登录验证，并将登录结果通过share memory发送给dds\_server\_mock和gtw\_server\_mock服务，这两个服务会通过信号机制等待接收用户账户和token，收到后立即向login server发起token验证。

### 性能测试模式

该模式下qb\_clients\_mock、dds\_server\_mock、gtw\_server\_mock等角色会向login server快速发起大量请求，并统计服务每秒返回的消息数。

# 接口数据准确性联合测试

## 测试方法

1. login server启动后没有任何用户数据，需要启动模拟数据源服务为其制造用户账户数据，在命令行模式启动user\_info\_supplier\_mock，命令为：

user\_info\_supplier\_mock.exe -a172.16.8.85 -c100

命令参数说明：

-a为msgbus server地址

-c为指定制造用户账户数据的条数

1. 命令行模式开启qb\_clients\_mock，命令为：

qb\_clients\_mock.exe -m1 -s240000 -i1 -w1 -a172.16.73.169 -p10080 -u0

说明：

-m为测试模式，0为测速模式，非0为验证模式，在验证模式下-s的值随意设置，其他参数为lava的接入参数。

1. 命令模式开启gtw\_server\_mock，命令为：

gtw\_server\_mock.exe -a172.16.8.85 -dgateway\_srv\_token\_verify.req.queue -lgateway\_srv\_token\_verify.req.queue.rspa -m1 -s16000

说明：

-a为msgbus server地址

-d为发送给目标队列的名称

-l为接收返回消息的队列名称

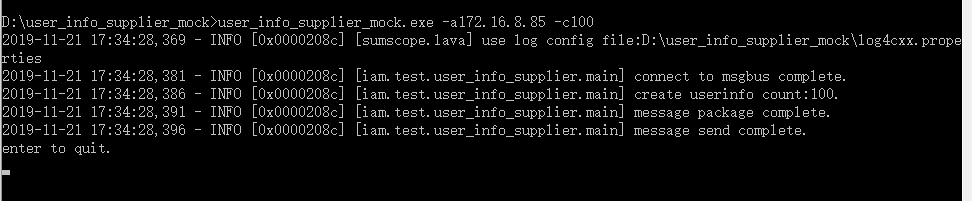
-m为测试模式，在验证模式下-s的值随意设置。

4. 命令行模式开启dds\_server\_mock，命令为：

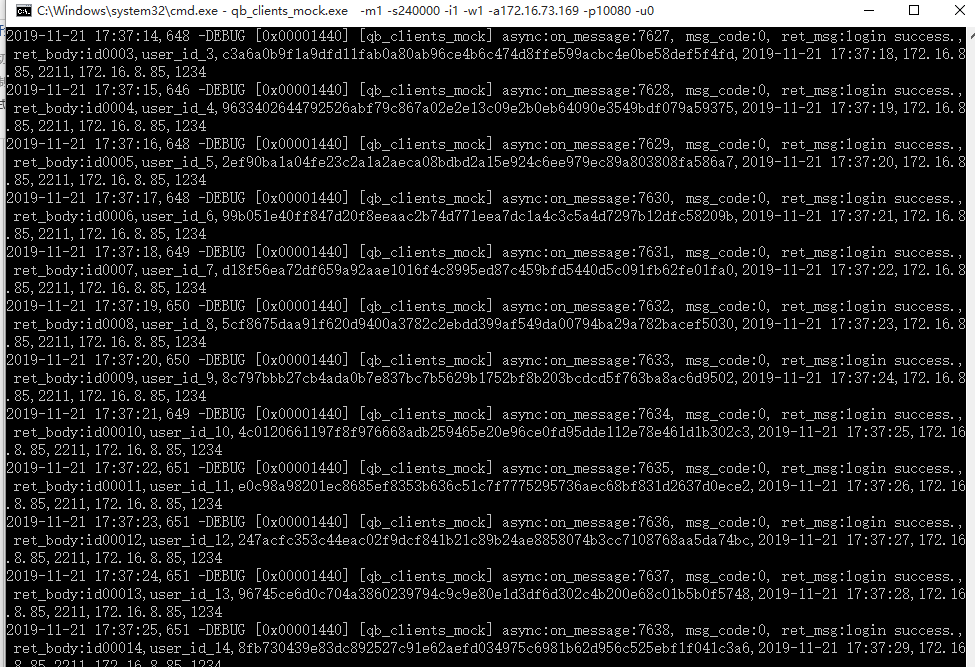
dds\_server\_mock.exe -m1 -s240000 -i1 -w1 -a172.16.73.169 -p10080 -u0

## 测试结果截图

### 模拟数据源执行结果



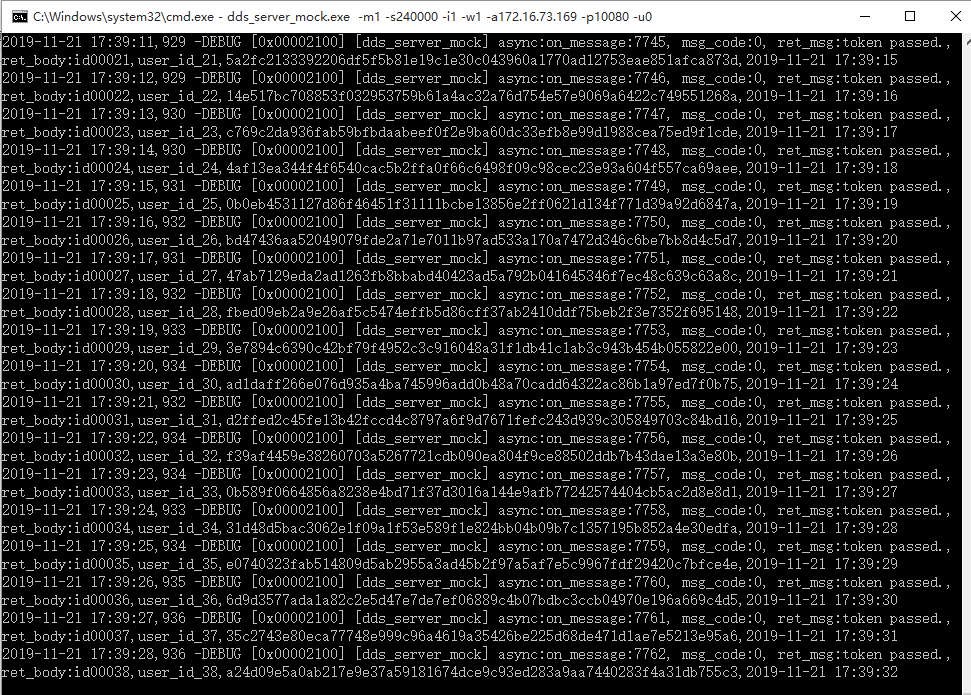
### 模拟QB客户端执行结果



### 模拟网关服务执行结果



### 模拟DDS服务执行结果



## 测试结论

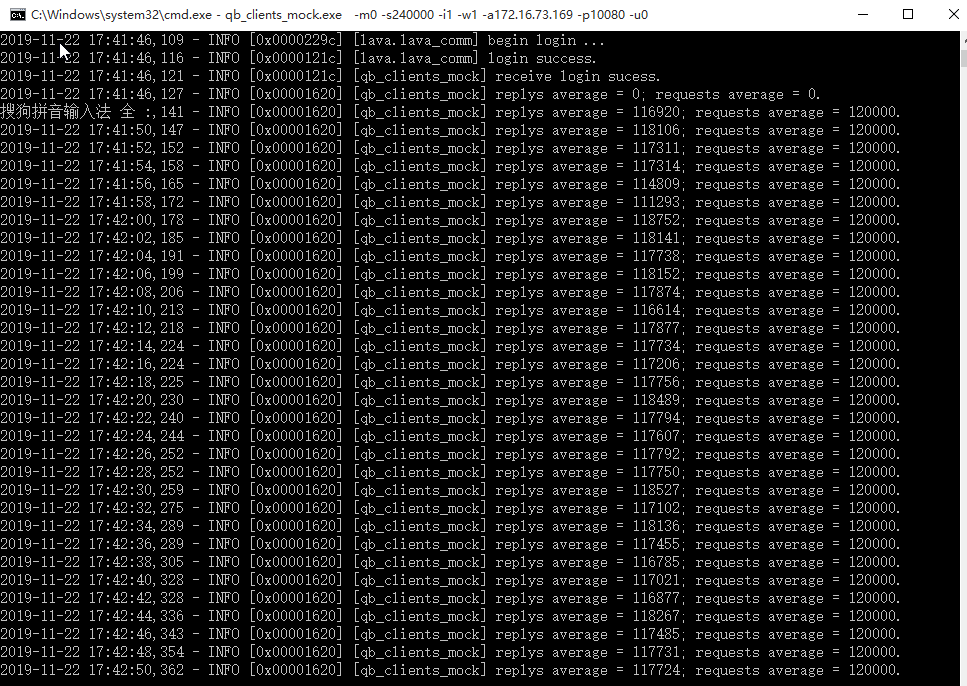
各个端口数据返回符合预期。

# 服务性能测试

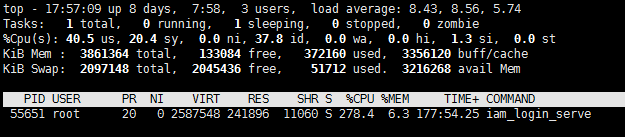
## lava box通信方式的性能测试

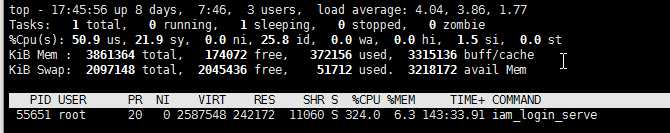
### 模拟QB客户端的性能测试

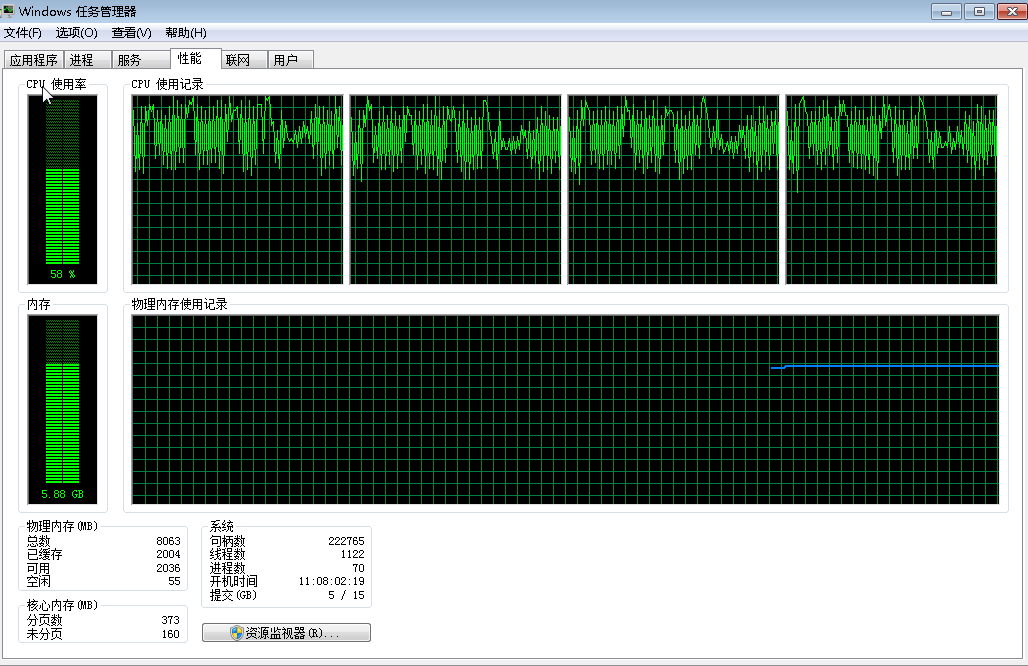
1. QB客户端限制并发访问量限制为每秒12万

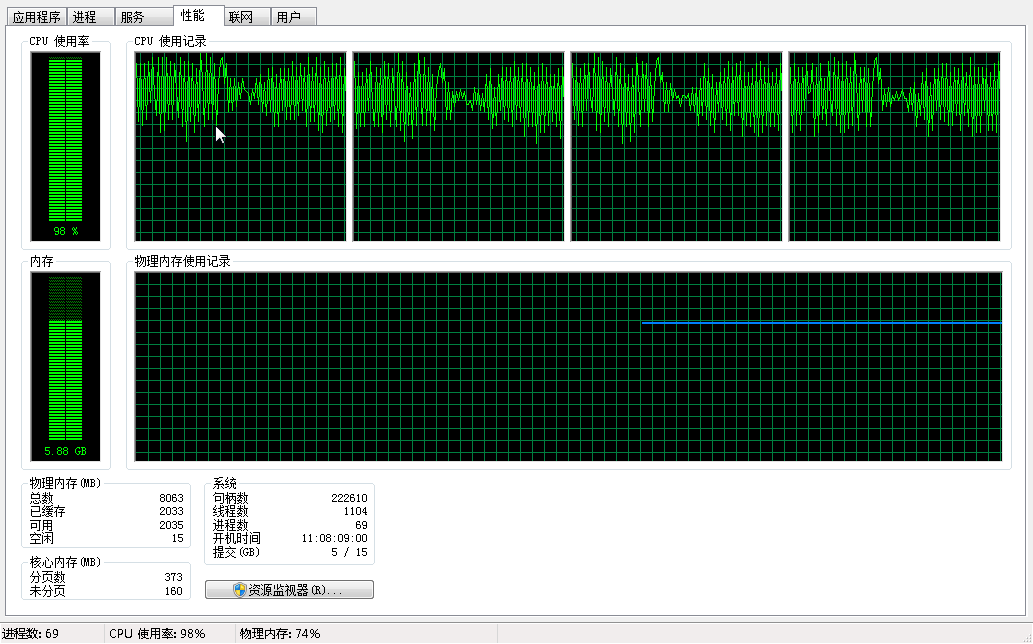


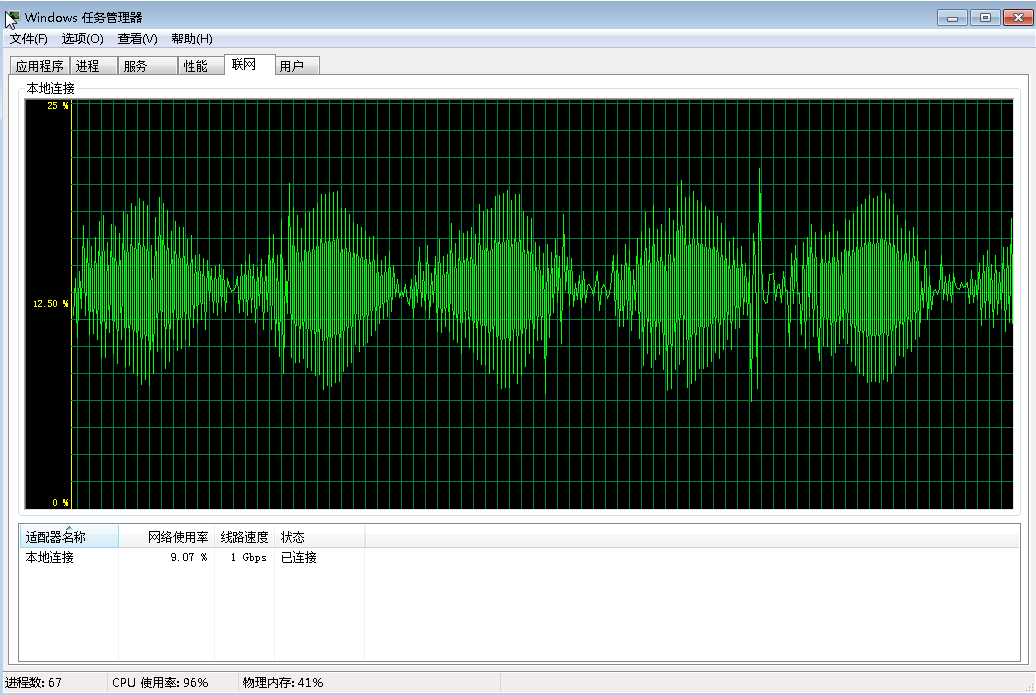
1. 服务端表现

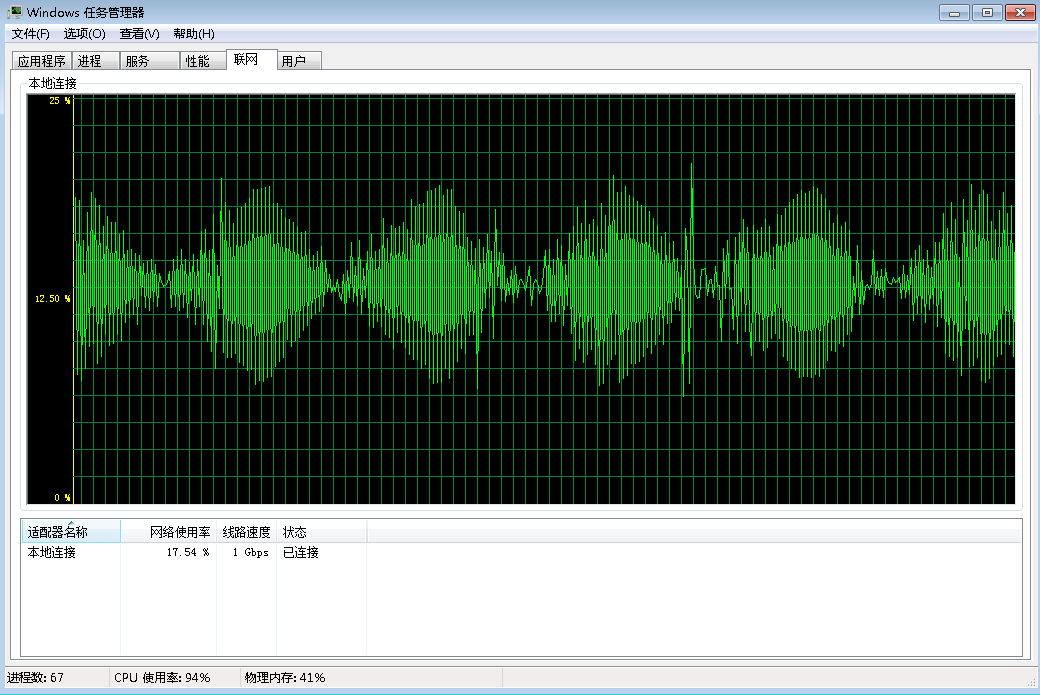




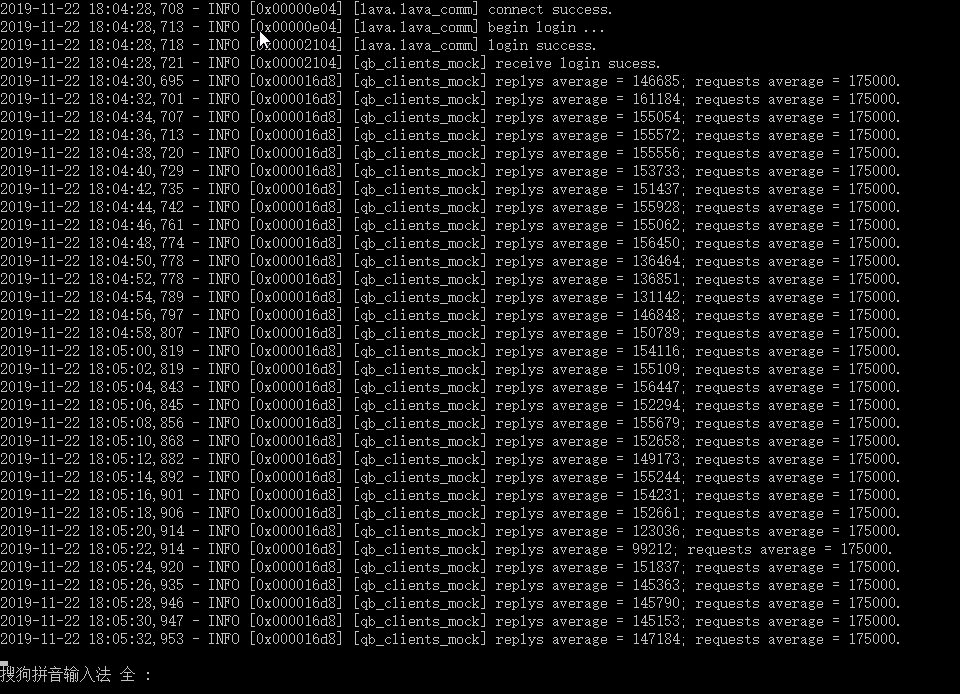




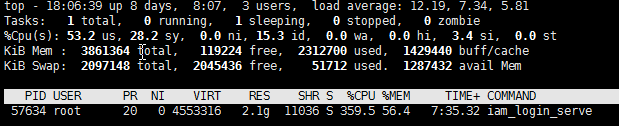




1. QB客户端并发访问量超过每秒12万

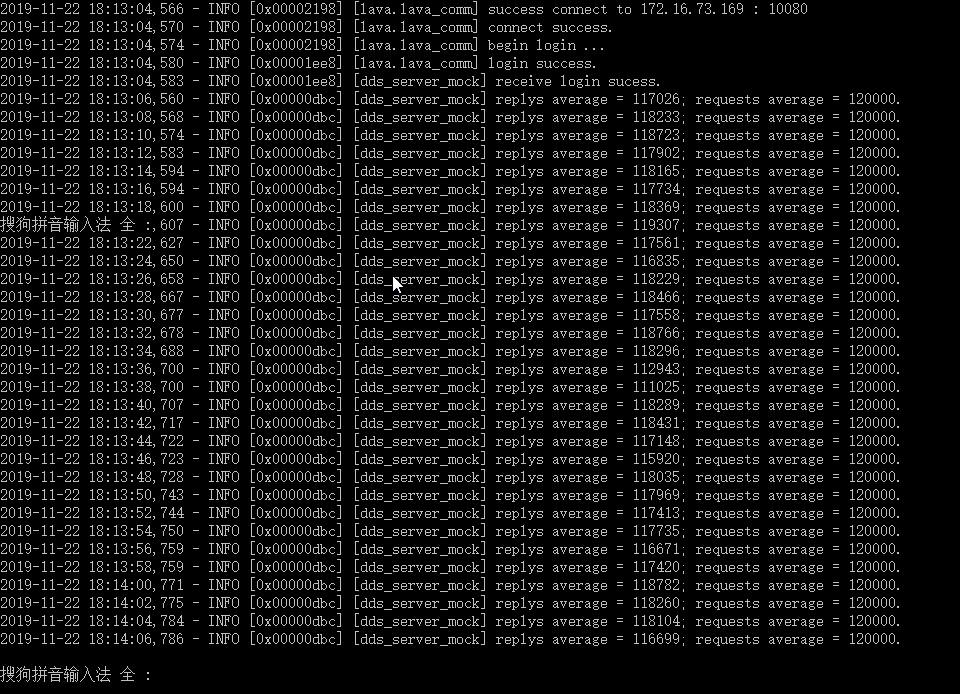


1. 服务端表现

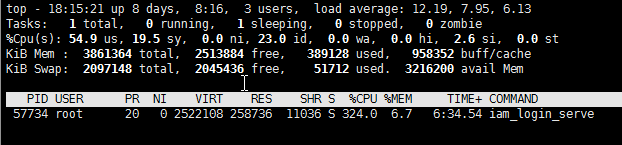


### 模拟QB客户端的性能测试

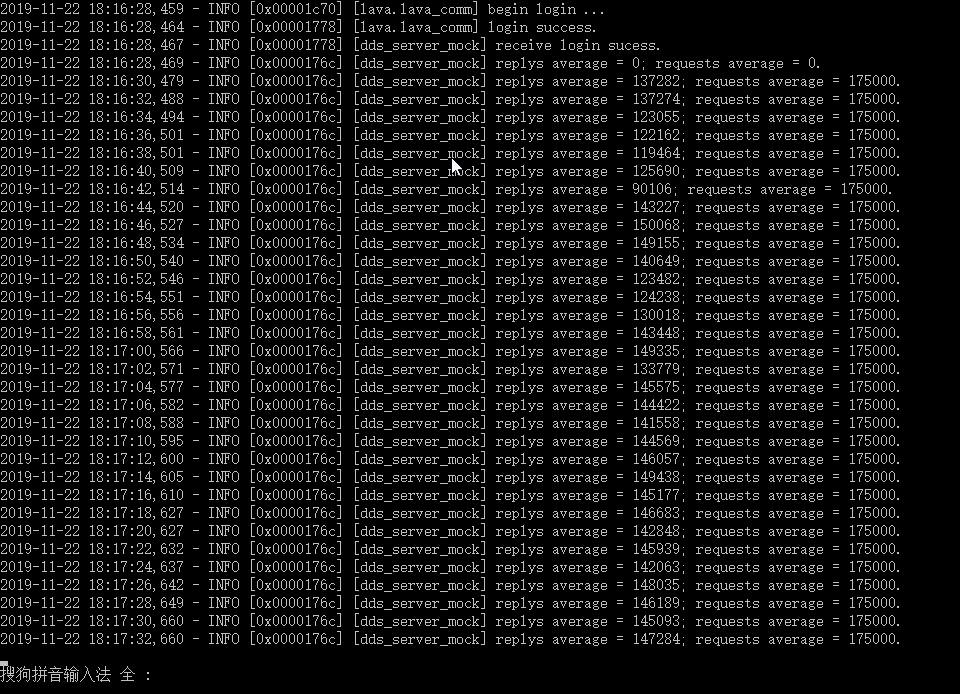
1. 限制访问量为每秒12万



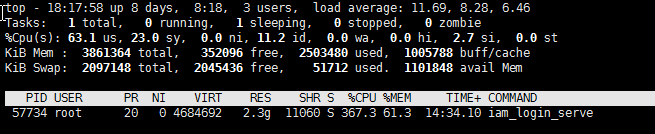
1. 服务端表现



1. DDS server访问量超过每秒12万



1. 服务端表现



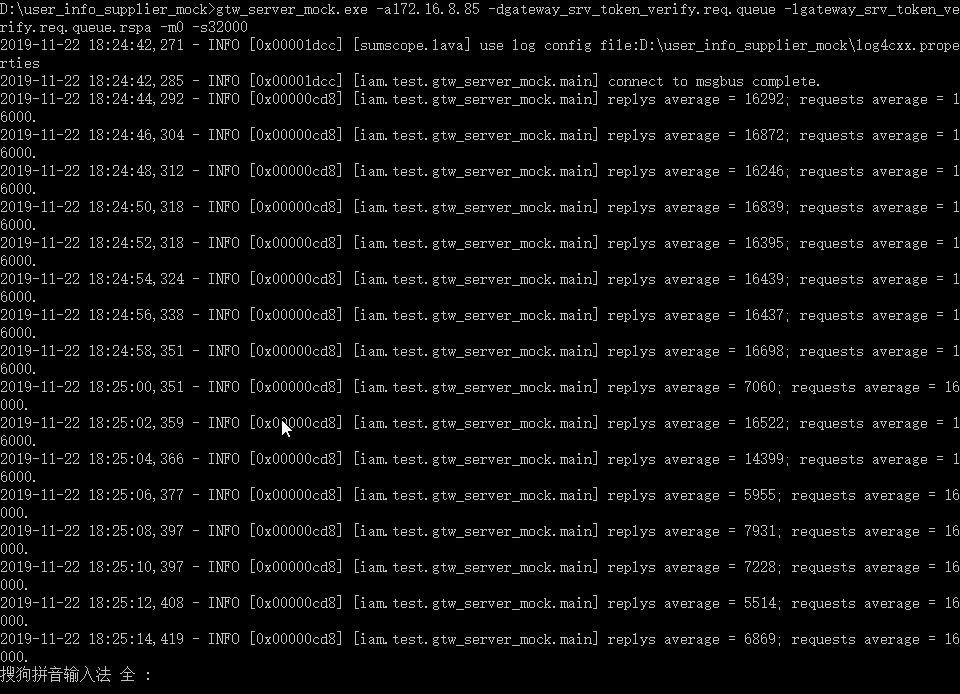
### 基于lava box业务性能测试结论

1. 从实验数据中可以看出QB客户端登录验证接口与DDS服务token验证接口在并发处理表现上基本一致，其吞吐量主要受lava box吞吐量的制约，吞吐量并没有因为两种接口的业务逻辑不同而差别很大，可见服务性能主要耗费在资源竞争、IO处理等方面，并没有被计算逻辑关系严重影响。
2. 基于lava box消息的处理吞吐量稳定在每秒12万次请求，小于或等于这个吞吐量，服务工作是稳定的，内存消耗在合理的范围内上下波动。
3. 当并发访问量高于每秒12万次时，服务处理速度峰值可突破15万次每秒，但是服务工作状态变得很不稳定，服务内存会快速上升，如果不及时降低访问压力，服务内存会最终耗尽。
4. 当压力超过12万次每秒时，持续一段时间，如果及时降低压力至合理范围内，则服务内存一段时间后会降至正常范围。
5. 目前很难做到高压力下服务内存稳定，因为即使把lava box消息不处理并抛弃掉，被抛弃掉的消息内存会由lava模块调用dereferance来释放，但是内存释放的速度仍然没有接收消息速度快。

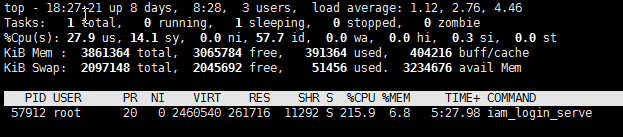
## msg bus通信方式的性能测试

### 模拟网关服务的性能测试

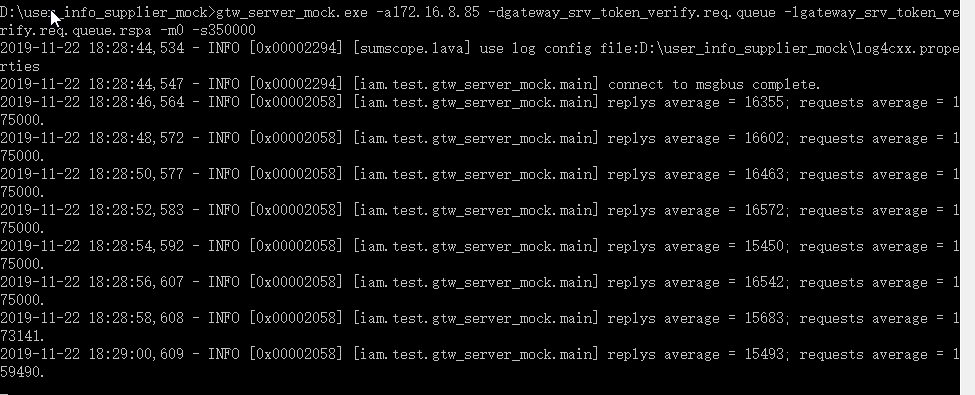
1. gateway server限制访问量为每秒16000条



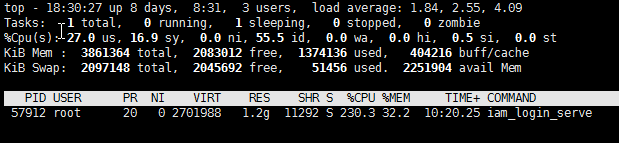
1. 服务端表现



1. gateway server突破访问量每秒16000条，大量发送



1. 服务端表现



### 基于msgbus业务性能测试结论

1. 基于msgbus的消息处理性能非常低下，从数据可以看出基本可以维持在1万6千次每秒的吞吐量。
2. 如果高于1万6千次每秒，则服务内存会迅速升高，如不降低压力，则服务内存会耗尽。
3. msgbus动态库中存在内存泄露，当服务吞吐量较大时，内存会逐渐增大，且有大量垃圾数据泄露在内存中无法释放。

# 单例服务运行稳定性测试

1. 经过优化和内存泄露修复等处理，服务稳定性能极大提高，一般的业务处理是很稳定的，如无网络断网等极端情况，服务运行时间达到7\*24小时是没有问题的。
2. 服务处理并发的能力稳定在每秒处理12万次左右的请求，在这个并发压力之下，服务内存消耗是稳定的。
3. 如果高于这个并发数，服务处理能力的峰值可能会达到15万次左右，但是服务内存会大量消耗，此时如果不降低并发强度，服务内存会不断增加直至耗尽。
4. 如果超过12万次并发数，持续一段时间后降低并发强度，则服务内存会慢慢进行回收，但是仍然有部分内存不能回收，原因目前还不清楚。
5. 已设计服务处理并发容限的阀值，可以通过配置文件来设定这个阀值，不同阀值的设定会提高或降低服务并发处理能力。

# 多实例服务数据同步验证

# 服务高可用测试