版本号：V1.0

**SecSpace客户端性能优化**

**阶段报告（500并发）**

刘大东

2018年09月14日

## 修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **修订日期** | **版本号** | **修订描述** | **修订人** |
| 2018.09.14 | V1.0 | 初稿 | 刘大东 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 目录

[一、概述 5](#_Toc524850636)

[1.1 报告简述 5](#_Toc524850637)

[1.2 优化目标 5](#_Toc524850638)

[1.3 术语说明 5](#_Toc524850639)

[1.4 参考文档 5](#_Toc524850640)

[二、测试环境部署 6](#_Toc524850641)

[2.1 服务端 6](#_Toc524850642)

[2.2 客户端 6](#_Toc524850643)

[2.3 网络拓扑图 6](#_Toc524850644)

[三、性能优化方案 6](#_Toc524850645)

[3.1 筛选测试 7](#_Toc524850646)

[3.2 瓶颈业务分析 7](#_Toc524850647)

[3.3 瓶颈接口分析 7](#_Toc524850648)

[3.4 接口优化 9](#_Toc524850649)

[3.5 迭代测试 10](#_Toc524850650)

[四、优化成果及相关结论 10](#_Toc524850651)

## 概述

### 1.1 报告简述

本报告是对SecSpace第13轮开发版本客服端性能优化阶段工作（500并发）的分析与总结。

在第12轮客户端性能摸底测试的基础上，根据第13轮开发的接口变动，讨论更新了性能测试业务流程，确定以登录、登录后解锁、仅解锁、获取应用列表、获取文件列表、获取策略和检查更新等7个性能测试业务，寻找满足500并发的服务器最低配置及各项配置参数和性能的相关性。

### 1.2 优化目标

通过各种技术手段，提升服务器性能，确认在500并发压力下，服务器要满足性能需求需要的最低配置，同时分析总结各配置参数对业务及接口性能的影响，确认系统瓶颈，为后续要求更高的性能优化工作提供有力依据。

### 1.3 术语说明

**瓶颈业务：**在规定的硬件配置下，不能满足性能需求的业务，该业务中的某个或某些接口对整个业务性能影响。

**瓶颈接口：**瓶颈业务中某个或某些性能表现较差，拉低了响应业务性能表现，甚至拉低整个系统性能表现的接口。

**响应时间：**应用系统从发出请求开始到客户端接收到所有数据所消耗的时间。

**并发用户数**：同一时刻与服务器进行数据交互的所有用户数量。

**吞吐率（Throughout）**：单位时间内从服务器返回的字节数。

**TPS（Transaction Per Second）**：表示服务器每秒处理的事务数。

**点击率**：每秒用户向服务器提交的请求数。

**资源利用率**：服务器系统中不同硬件资源被使用的程度。

**思考时间**：用户在进行操作时，每个请求之间的时间间隔。

### 1.4 参考文档

《SecSpace客户端性能测试报告-刘大东》（第12轮开发版）

## 测试环境部署

### 2.1 服务端

Nginx服务器

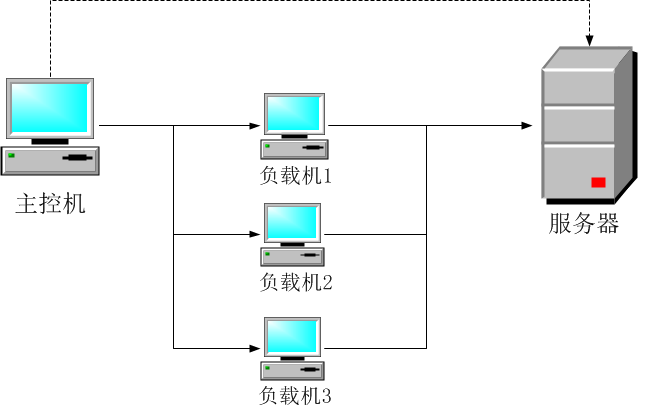
|  |  |
| --- | --- |
| 操作系统 | Linux 3.10.0-693.el7.x86\_64 CentOS Linux release 7.4.1708(Core) |
| 虚拟机版本 | 11 |
| Product Name | 440BX Desktop Reference Platform（虚拟机） |
| CPU | Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz |
| 核数 | 8,1 |
| RAM | 8G,16G |
| Web服务 | nginx |
| IP | 192.168.122.1 |

### 2.2 客户端

主控机+负载机×3

|  |  |
| --- | --- |
| 操作系统 | win7 64 |
| 虚拟机版本 | 11 |
| Product Name | 440BX Desktop Reference Platform（虚拟机） |
| CPU | Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz |
| 核数 | 16 |
| RAM | 32G |
| IP | 192.168.3.112（113,114,115） |

### 2.3 网络拓扑图



主控机和负载机均完全安装LoadRunner 11，负载机受控于主控机，分担负载。

## 性能优化方案

性能优化主要分为筛选测试、瓶颈业务分析、瓶颈接口分析、接口优化、迭代测试三个部分。

### 3.1 筛选测试

1）场景设计

保持服务器配置不变（5进程8核8G服务器），以步长50/5min渐进至500并发为比对场景，分别运行被测业务对应的脚本，根据各业务的性能需求（允许的最大响应时间），判定对应业务是否已经满足性能需求，并获取不满足要求的业务的需求并发。

2）测试结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **业务** | **脚本** | **需求并发** | **最大并发** | **响应时间s** | **性能需求s** | **结论** |
| 登录 | v13\_login | 400 | 500 | 3.7 | 3 | 不满足需求 |
| 登录后解锁 | v13\_unlock | 300 | 500 | 5.3 | 3 | 不满足需求 |
| 仅解锁 | v13\_unlockonly | 250 | 500 | 6 | 3 | 不满足需求 |
| 获取应用列表 | v13\_getapps | —— | 500 | 0.9 | 1 | 满足需求 |
| 获取文件列表 | v13\_getfiles | —— | 500 | 0.9 | 1 | 满足需求 |
| 获取策略 | v13\_getpolicy | —— | 500 | 1.8 | 3 | 满足需求 |
| 检查更新 | v13\_checkupdate | —— | 500 | 0.48 | 0.5 | 满足需求 |

由上表可知，优化前Secspace的瓶颈业务分别有登录、登录后解锁、仅解锁三个，其余业务在500并发下，均满足性能需求。

### 3.2 瓶颈业务分析

下表为瓶颈业务登录、登录后解锁、仅解锁在各自需求并发下的测试结果摘要。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **脚本** | **服务器配置** | **需求并发** | **响应时间s** | **CPU占用%** | **内存占用%** |
| v13\_login | 5进程8核8G | 400 | 2.9 | 78.4 | 93.8 |
| v13\_unlock | 5进程8核8G | 300 | 3.2 | 80.6 | 93.9 |
| v13\_unlockonly | 5进程8核8G | 250 | 2.8 | 82.7 | 92.9 |

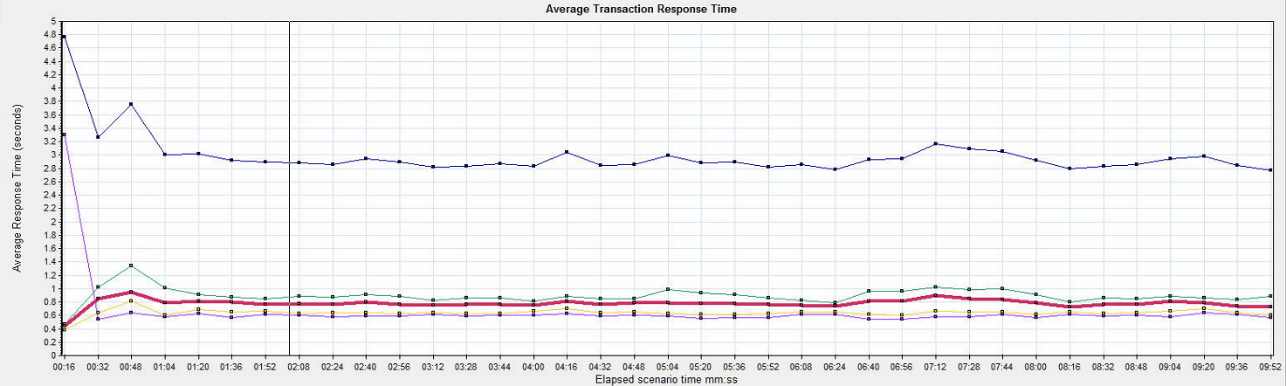
由上表可知：登录（v13\_login）、登录后解锁（v13\_unlock）、解锁（v13\_unlockonly）三个瓶颈业务对应的需求并发分别为400、300、250，CPU占用均在80%左右，内存占用均超过90%，说明8G内存不够用，或者存在不良代码过度占用内存。

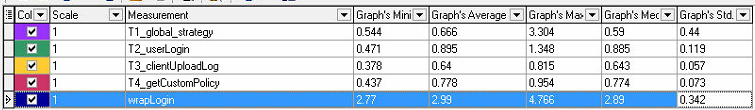
### 3.3 瓶颈接口分析

分析瓶颈业务中性能表现较差的接口。

1. 登录（v13\_login）

登录业务包含获取全局配置（global\_strategy）、用户登录（userLogin）、上传登录日志（clientUploadLog）、获取客户端策略（getCustomPolicy）四个接口。下图为登录业务响应时间曲线：

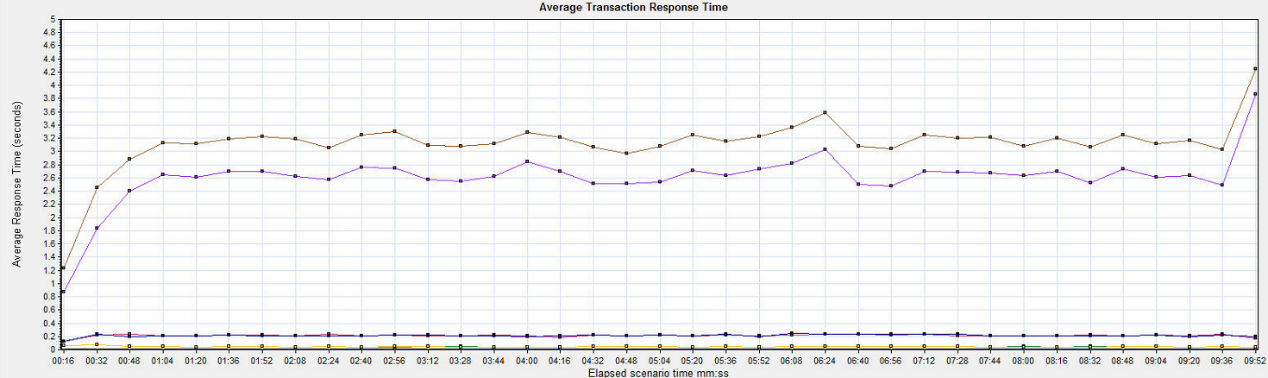


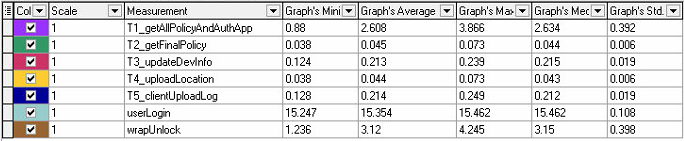


由上图可知，登录业务所包含的四个接口响应时间相当，其中接口userLogin和getCustomPolicy所占时长较高。

1. 登录后解锁（v13\_unlcok）

登录后解锁业务包含获取全部策略和授权应用（getAllPolicyAndAuthApp）、上传最终应用策略（getFinalPolicy）、上传设备信息（updateDevInfo）、上传位置信息（uploadLocation）、上传解锁日志（clientUploadLog）五个接口。下图为登录后解锁业务响应时间曲线：

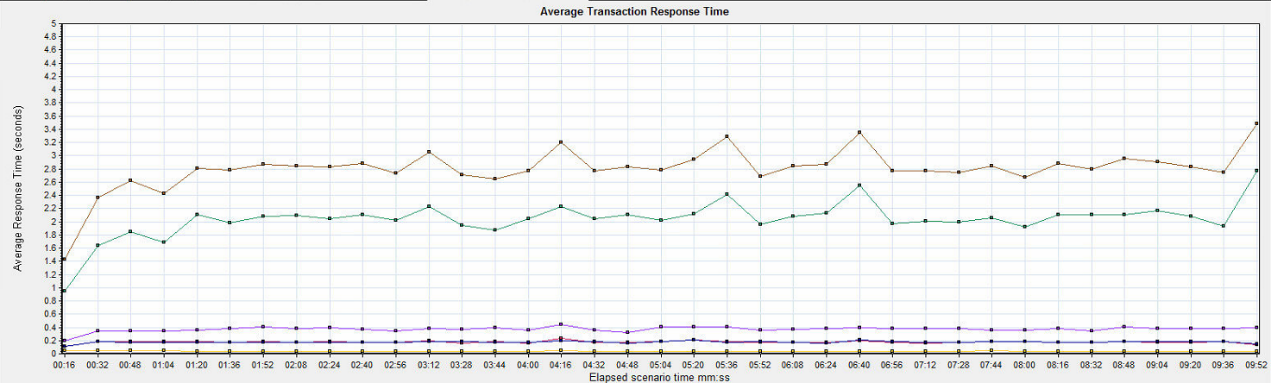


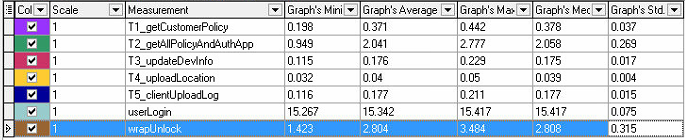


由上图可知，登录后解锁业务中获取全局策略和授权应用接口（getAllPolicyAndAuthApp）响应时间占比远大于其他接口，响应时间曲线走势几乎和全业务响应时间一致，呈现强相关，这说明对于该接口的优化可以更明显的提升整个业务的性能。

1. 仅解锁（v13\_unlockonly）

仅解锁业务包含获取客户端策略（getCustomPolicy）、获取全部策略和授权应用（getAllPolicyAndAuthApp）、上传设备信息（updateDevInfo）、上传位置信息（uploadLocation）、上传解锁日志（clientUploadLog）五个接口。下图为仅解锁业务响应时间曲线：





由上图可知，仅解锁业务中获取全局策略和授权应用接口（getAllPolicyAndAuthApp）响应时间占比远大于其他接口，响应时间曲线走势几乎和全业务响应时间一致，呈现强相关，这说明对于该接口的优化可以更明显的提升整个业务的性能。

### 3.4 接口优化

#### 3.4.1 接口定位

根据筛选测试结果，组织了一次性能分析会，结果显示登录和解锁不符合性能需求；根据瓶颈分析结果，最耗时的接口为用户登录接口(/p/user/userLogin)和获取所有策略接口(/p/user/getAllPolicyAndAuthApp)，较耗时的接口为获取客户端策略接口(/p/user/getCustomerPolicy)。

#### 3.4.2 接口优化

1）梳理代码逻辑

对定位到的耗时接口进行代码逻辑梳理，挖掘优化点

2）优化登录接口(/p/user/userLogin)

客户端带着自身标识，用户名，密码进行登录认证时，在设备与用户进行绑定的过程中，将对数据库的插入／更新操作合并成一次数据库查询，即达到目的: 当数据库中存在该条记录，实施update操作；不存在时，实施insert操作

3）优化获取所有策略及授权应用接口(/p/user/getAllPolicyAndAuthApp)

查询计数count()方法: 由于python的mongodb驱动的原因，不提供count()方法中添加查询条件；

to\_list(length)方法：数据库查询结果需使用此方法转换结果，此方法不会提前申请内存。

4）优化获取客户端策略接口(/p/user/getCustomerPolicy)

更改原有代码逻辑，将计算处理的部分实现异步，不阻塞主进程；并减少查库次数。

### 3.5 迭代测试

对于接口的每一次优化，采用特定的场景（5进程8核8/16G、500并发10min）测试接口优化对相应业务的优化效果。下表为该阶段迭代最终优化效果：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **脚本** | **服务端配置** | **并发** | **CPU占用%** | **内存占用%** | **响应时间s** | **需求并发** |
| 1 | v13\_login | 5进程8核8G | 500 | 79.90 | 45.14 | 2.90 | 500 |
| 2 | v13\_unlock | 5进程8核8G | 500 | 82.60 | 43.61 | 5.00 | 350 |
| 3 | v13\_unlockonly | 5进程8核8G | 500 | 74.80 | 41.99 | 5.50 | 300 |
| 4 | v13\_login | 5进程8核16G | 500 | 80.20 | 13.03 | 2.90 | 500 |
| 5 | v13\_unlock | 5进程8核16G | 500 | 83.50 | 22.29 | 5.00 | 350 |
| 6 | v13\_unlockonly | 5进程8核16G | 500 | 76.10 | 24.72 | 5.40 | 300 |

由上表可知，当设置5个进程，服务器配以8核CPU、8G内存时，CPU占用约为80%，内存占用约为43%。再将内存增至16G，CPU占用几乎不变，内存占用率则腰斩，折算之后，实际占用的内存几乎没有变化，同时响应时间也几乎没有变化。这说明在5进程8核8G的服务器配置下，增加内存已不能明显提升业务性能。

## 优化成果及相关结论

本次性能优化成果及相关结论如下：

1. secspace系统优化之前，在5进程8核8G的服务器配置500并发压力下，存在登录、登录后解锁、仅解锁三个瓶颈业务，业务响应时间分别为3.7s、5.3s、6s，需求并发分别为400、300、250，CPU占用约80%，内存占用约93%。
2. 通过将登陆业务相关接口的优化，在5进程8核8G的服务器配置500并发压力下，将登录业务的响应时间由3.7s降至2.9s，满足3s响应需求。优化方法如下：
3. 将上传登录日志接口合并到用户登录接口，减少登录业务请求次数；
4. 在设备与用户进行绑定的过程中，将对数据库的插入和更新操作合并成一次数据库查询，减少数据库查询次数。
5. 更改获取客户端策略接口(/p/user/getCustomerPolicy)原有代码逻辑，将计算处理的部分实现异步，不阻塞主进程，并减少查库次数。
6. 通过对代码的梳理，登录、登录后解锁、仅解锁三个业务的内存占用都明显降低，在5进程8核8G的服务器配置500并发压力下，内存占用率由93%左右降至43%左右。
7. 在5进程8核8G的服务器配置500并发压力下，通过对解锁业务相关接口的优化，登录后解锁响应时间由5.3s降至5s，需求并发由300提升至350，仅解锁响应时间由6s降至5.5s，需求并发由250提升至300，优化效果不是很明显。
8. 获取全局策略和授权应用接口（/p/user/getAllPolicyAndAuthApp）是解锁业务的瓶颈接口，优化该接口可以明显提升解锁业务性能。经讨论，唯有改变原有代码逻辑，引入缓存机制，充分利用内存，减少数据库查询次数，才能明显提升该接口的性能，进而提升解锁业务性能，拉高整个系统支持的并发。
9. 考虑到单服务器CPU核数通常不超过8核，目前的测试都是在5进程8核的约束条件下进行的，所以优化前后CPU占用几乎都在80%左右。经讨论，下一阶段放开这一约束，通过改变进程数和CPU核数探寻这两项设置与整体性能的相关性。由于这种相关性会受获取全局策略和授权应用接口的影响，尤其是将要引入的缓存机制，会明显影响系统对CPU和内存的依赖比例。所以进程数和CPU核数对系统性能的精细研究，最好放到引入缓存机制之后进行。