**自动化质量测试平台概要设计方案**

**目录**

[1 术语&缩略语 2](#_Toc31380)

[2 原始需求及需求分析 2](#_Toc23891)

[3 实现方案 2](#_Toc7068)

[3.1 整体架构 2](#_Toc10339)

[3.1.1 Test server 功能 3](#_Toc31129)

[3.1.2 Bench server功能 4](#_Toc29695)

[3.1.3 Waf server功能 5](#_Toc31911)

[3.2 处理流程图 6](#_Toc30037)

[3.2.1 发送压测指令 6](#_Toc9418)

[3.2.2 Waf服务器接收压测命令 7](#_Toc5534)

[3.3 数据接口 8](#_Toc8826)

[3.3.1 waf server脚本模块化： 8](#_Toc1803)

[4 非功能需求 9](#_Toc31830)

[5 遗留问题 9](#_Toc9776)

[6 附录 9](#_Toc29541)

# 术语&缩略语

|  |  |
| --- | --- |
| 名词 | 说明 |
|  |  |

# 原始需求及需求分析

目标：根据不同压测模式，对waf性能及质量进行分析，保证waf可用性及稳定性。

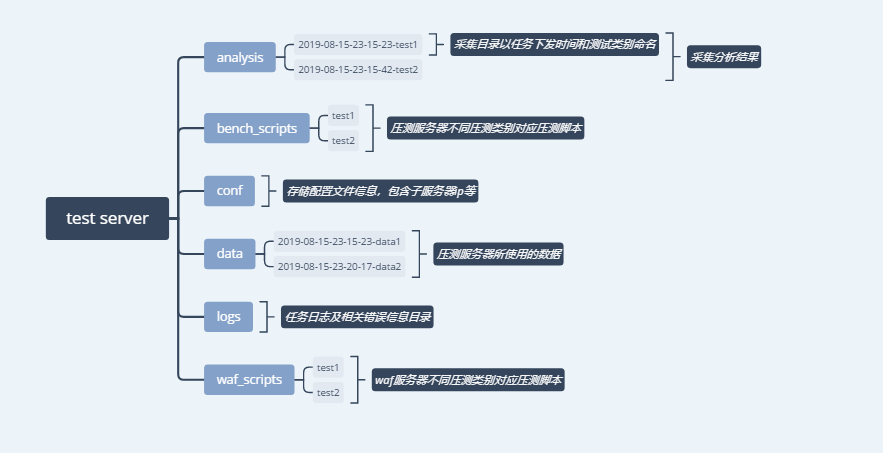
# 实现方案

## 整体架构



### Test server 功能

目录结构：



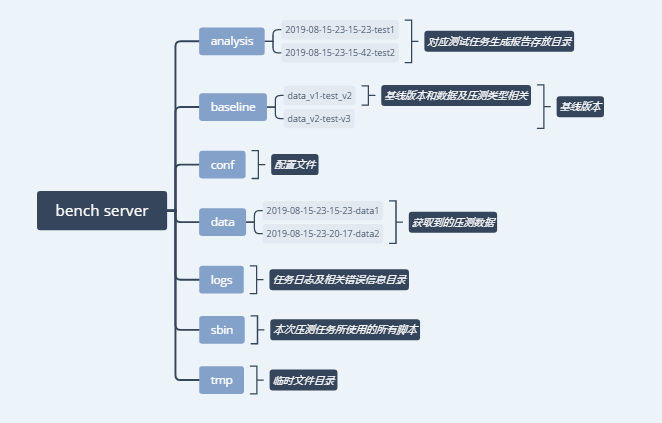
1、发送压测请求到各个压测服务器和waf服务器

2、接收压测统计结果

3、发送脚本文件和压测数据

### Bench server功能

目录结构：



1、接收压测命令

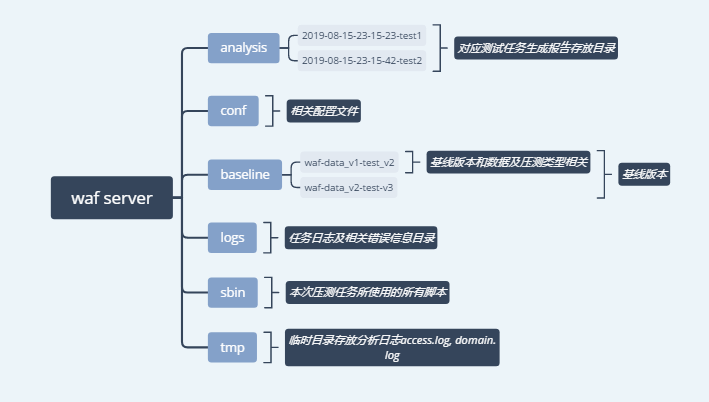
2、获取更新压测脚本，压测数据

3、分析对比压测结果

4、上报压测结果

### Waf server功能

目录结构：



1、接收压测命令

2、获取更新压测相关内容

3、分析对比结果

4、上报压测结果

Conf文件夹： server.config : 一行一个配置

例如： server\_host www.test\_server.com:60000

server\_ip 180.101.154.91

server\_add\_header field value

## 处理流程图

### 发送压测指令



1. 用户发送压测请求命令到test server
2. Test server 发送准备压测请求到压测服务器和waf服务器
3. 压测服务器和waf服务器根据唯一id从test server下载打包文件（打包文件包含rpm，脚本，数据等内容），下载完成响应ok
4. Test server 发送正式压测请求到waf服务器和压测服务器
5. 压测服务器开始压测，waf服务器开始实时分析
6. 压测服务器请求结束，通知test server, test server发送请求到waf服务器进行结果收集

### Waf服务器接收压测命令



1. test server 发送 /pre\_bench请求到waf服务器
2. waf服务器下载响应压缩文件，解压到sbin目录，启动prepare.sh
3. 压测前预处理
4. 响应对应内容

## 数据接口

### waf server脚本模块化：

执行脚本：

prepare.sh

start.sh

stop.sh

finish.sh

其它中间使用脚本

控制模块接收到压测命令时，调用start.sh

当压测结束时，调用finish.sh脚本

prepare.sh: 开始压测前操作，如修改部分文件，设置目录等

start.sh： 调用脚本，产生数据

finish.sh: 结束时操作，分析，打包分析结果，调用上传命令等

stop.sh: 终止当前压测操作

从test server下载tar.gz包

http:// xxx.xxx.xxx.xx:xxxx/download/waf/guid.tgz

start.sh接收参数test\_type baseline\_version waf\_version data\_version operation guid

test\_type:压测类型，质量测试，压测，单域名，单规则

baseline\_version：基线版本对应waf版本

waf\_version: waf版本

data\_version: 压测数据版本

operation: 操作类型：0：测试基线版本 1：测试waf版本

finish.sh: 对应文件进行采集，打包，并发送到指定端口

curl -x POST http://127.0.0.1:60001/finished\_bench -H "content-type:application/json" -d info.json

info.json:

{

"result":"success", /\*执行成功\*/

"file\_path":"/usr/local/..." , /\*打包文件地址\*/

"task\_id": "efa-1234-adfi234"

}

# 非功能需求

# 遗留问题

# 附录

无