

# WEB进程管理系统

项目需求：

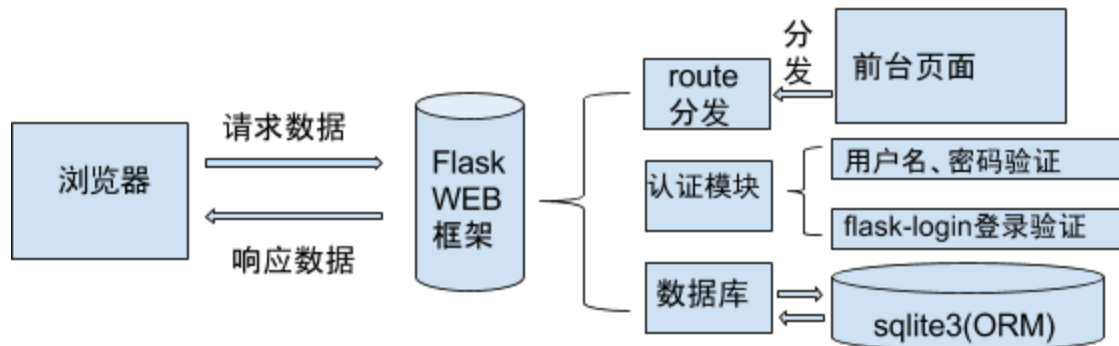
实现一个BS架构的进程管理系统，完成对linux宿主服务器的系统进程进行实时查看，获取当前进程情况，和远程对某些特定的服务进程进行停止或启动操作等。无须远程登录服务器，通过web页面即可查看系统信息和管理系统进程。

词汇表：

术语	解释
flask	Flask 是一个 Python 实现的 Web 开发微框架。
ORM	是一种程序设计技术，用于实现面向对象编程语言里不同类型系统的数据之间的转换。从效果上说，它其实是创建了一个可在编程语言里使用的“虚拟对象数据库”。

1.web框架:flask

系统结构：



数据库:ORM,后期扩展的话再说。

需用到的额外flask模块:flask-login, flask-wtf, flask-sqlalchemy 等

2.功能模块

2.1 功能分为三个模块，第一个模块展示主机的基本状态信息。

(1) 页面表现形式：以列表形式展示当前主机信息(信息无需做存储处理)

系统版本	直接调用后台函数获取
主机名	直接调用后台函数获取
其他信息	直接调用后台函数获取

(2) 后台表现形式：利用python代码执行系统命令获取系统信息，执行结果通过json数据格式返回，再将数据格式化处理后通过web框架返回到前台显示。

```
ret = {
    "is_exec": False,
    "cmdret": cmdout,    # 命令错误信息
    "retopt": "Unkown"  # 结果替代信息
}
```

2.2 第二个模块，根据当前已经运行的服务进程信息，对主机进程进行管理(实现常见服务的启动和停止)

(1) 页面表现形式：以table形式列出当前主机资源占用情况，动态显示以定时刷新数据形式展示(仅提供动态显示功能)。

系统任务情况	220 total, 1 running, 219 sleeping, 0 stopped, 0 zombie
CPU 占用情况	0.2 us, 0.0 sy, 0.0 ni, 99.5 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st
内存占用情况	7899824 total, 5210804 free, 761824 used, 1927196 buff/cache
swap占用情况	7995388 total, 7995388 free, 0 used. 5827944 avail Mem

显示用户刷新页面时的服务信息(TCP和UDP服务)，可根据服务名称查询(后台查询)，管理员以表单形式提交需要启动和停止的服务进程。

前台服务进程显示：

服务协议	本地服务地址和端口	外部服务地址和端口	服务状态	PID号/程序名称
tcp	0 0.0.0.0:111	0.0.0.0:*	LISTEN	1/systemd

表单输入查询：

输入服务名称查询	后台在进程资源中模糊匹配要查询的服务名以list形式显示详细信息
----------	----------------------------------

进程管理表单提交：

服务名(须全名)：	例如 mysql
操作类型(单选按钮)：	可选 < 启动 停止 重启 >
运行自定义命令(可选)：	例如：supervisorctl -c /etc/supervisord.conf restart all

具体要填写哪些字段信息视情况而定。不可操作或操作有误的进程操作指令，以提示框的信息反馈给管理员。

(2) 后台表现形式：利用python代码执行系统命令。

示例代码：

```
def cmd_run(args):  
    is_exec = False  
    p = subprocess.Popen(args, stdout=subprocess.PIPE, stderr=subprocess.PIPE)  
    (stdout, stderr) = p.communicate()  
    if stdout:  
        is_exec = True  
        return is_exec, stdout  
    return is_exec, stderr
```

(3) 关于实时性问题：

目前先使用前台页面定时请求数据的方法(动态显示)，后期有需要再学习server push技术实现推送。

2.3 第三个模块，列出管理员在web页面对主机服务进程管理的操作日志。

(1) 页面表现形式：以"徽章"和列表形式向管理员提醒和展示操作日志记录

提醒方式	新记录“徽章”显示
记录显示	数据库查询以列表形式显示

(2) 后台处理形式：将用户在第二个模块里所做的增杀进程日志插入数据库，提供日志查询功能。  
操作日志数据库表：

日志时间(logTime)	操作进程时记录的日期
事件类型(logType)	进程管理行为记录

结果状态(retStatus)	进程操作成功还是失败
查看状态(logStatus)	标志此条消息是否已被查看

2.4 前台与后台数据交互：使用web框架模版和ajax，数据格式为json

3. 已有功能：

(1) 登录验证，实现了管理员用户名和密码登录验证

登录页面展示

前台操作	键入用户名和密码。
后台操作	用户身份认证：使用flask-wtf, flask-login 模块登录认证

(2) 实现了页面基本布局，可展示当前服务器的基本状态信息，例如操作系统类型，主机名，定时任务等。