# 软件需求规格说明

一、概要

**软件系统必须具备的功能及性能、其特征和必须遵守的约束**

黄色的部分是现有MANO不具备的功能！因而我们要加功能，而不是修改功能！

必备功能：

1）根据SFC请求，将所需的VNF部署到服务器上，用户的流量从服务器发送出后必须按SFC的顺序经过每种VNF，最后送达目的主机

2）数据中心单节点故障（除了用户所在主机）不影响用户的正常业务

3）用户流量变大时，可以采用垂直/横向扩展承载增大的流量

4）首要VNF所在节点故障后，可以自动将首要VNF迁移到指定节点

5）可以实时查看每个SFC的状态信息和SLO

必要性能：

1）单节点故障保护率100%

2）垂直扩展时间低于1s

3）横向扩展时间低于10s

4）人为蓄意扰动抑制成功率>95%

**估算日程安排、工作量和资源**

NFV管理和编排系统可以分为前端（提交SFC编排请求、查看SFC服务等级指标、提交SFC删除请求）、后端（SFC编排、VNF库维护、SFC删除、SFC自适应扩缩容/故障恢复）、网络监控控制（获取VNF/server、转发路径、网络拓扑信息，维护DCN信息库、编写SDN控制器应用）三大部分：

1）前端部分需要自定义好需要展示的界面信息(可以参考Openstack Tacker的代码进行设计和实施)。定义VNF库和DCN数据库中的数据格式后，工作量预估在 1k LoC。

2）后端部分自己开发，工作量预估在5k LoC（有现实参考，有人自己用python写过调用docker接口实现CRUD功能，代码量不到3k LoC）。（在tacker的基础上增加/修改代码的可行性较差，因为目前的tacker还很不稳定，安装过程遇到很多bug；而且Tacker默认支持的SDN控制器是opendaylight，太重量级了比较难上手；不如自己开发MANO）

3）网络监控控制部分自己开发，工作量预估在1k LoC。（可以利用K8s的资源监控接口）

**分工**

需要安排2个人，前端+网络监控控制部分由一个人实现；后端部分由一个人实现；

**明确开发的产品**

NFV管理和编排系统

**明确交付的产品**

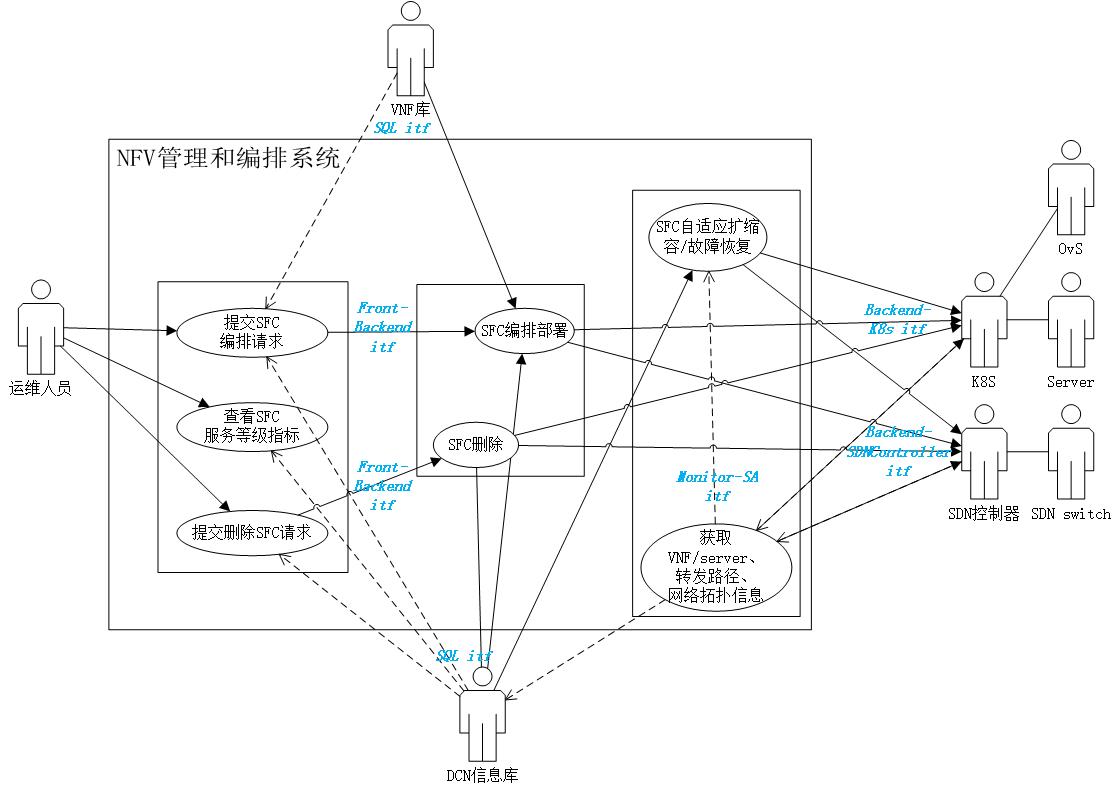
NFV管理和编排系统

**指定培训教材和培训计划**

K8s权威指南

Ryu官网指南

二、接口



1. **Front-Backend interfaces**

前端与后端的接口

利用rabbitMQ实现

接口格式定义：

1. **Monitor-SA interfaces**

网络监控与控制的接口

接口格式定义：

1. **Backend-K8s interfaces**

后端与K8s的接口

1. **Backend-SDN\_Controller interfaces**

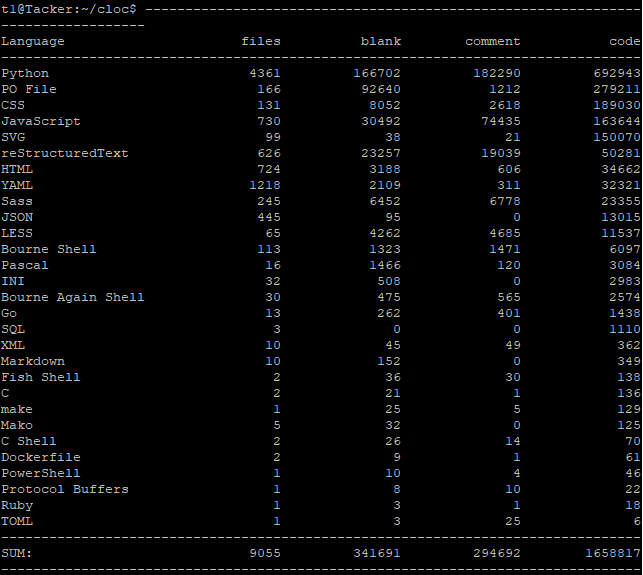
后端与SDN控制器的接口

1. **SQL interface**

各种模块与VNF和DCN信息库的接口

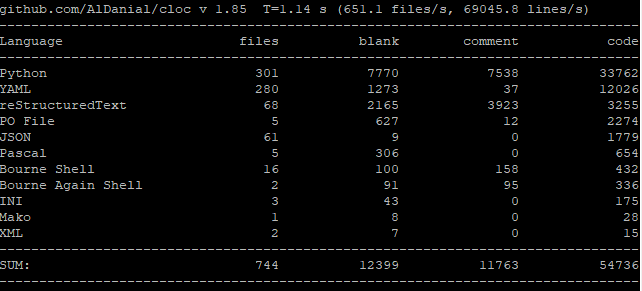
附录一 Tacker代码量

Tacker最小化安装，即tacker+所需的组件（比如horizon、keystone、networking-sfc、kuryr-k8s、tacker-horizon）：需要700k LoC Python

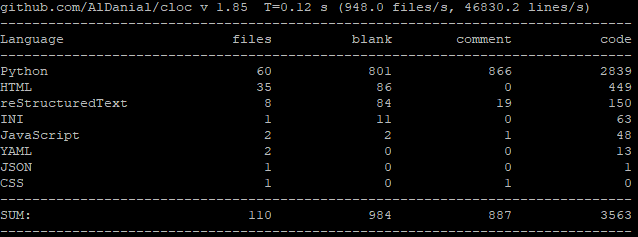


具体统计数据：

1）tacker文件夹：33k LoC Python、12k LoC YAML



2）tacker-horizon文件夹：3k LoC Python、12k LoC YAML



3）networking-sfc文件夹：28k LoC Python、12k LoC YAML

