**基于网络演化模型的云化虚拟化网络可靠性评估软件概要设计说明文档**

北京航空航天大学

可靠性与系统工程学院

电话：82339031

传真：82339031

地址：北京市海淀区学院路37号

电子邮件：hn@buaa.edu.cn

2020年12月

目录

[第一章 引言 1](#_Toc59786010)

[1.1 编写目的 1](#_Toc59786011)

[1.2 背景 1](#_Toc59786012)

[1.3 定义 2](#_Toc59786013)

[1.4 参考资料 3](#_Toc59786014)

[第二章 总体设计 4](#_Toc59786015)

[2.1 需求规定 4](#_Toc59786016)

[2.2 运行环境 4](#_Toc59786017)

[2.3 开发环境 4](#_Toc59786018)

[2.3.1 硬件环境 4](#_Toc59786019)

[2.3.2 软件环境 4](#_Toc59786020)

[2.3.3 开发语言 4](#_Toc59786021)

[2.4 基本设计概念 5](#_Toc59786022)

[2.4.1 单业务可用度 5](#_Toc59786023)

[2.4.2 整网业务可用度 5](#_Toc59786024)

[2.5 功能结构设计 5](#_Toc59786025)

[第三章 数据结构设计 7](#_Toc59786026)

[3.1 网络演化对象数据结构设计 7](#_Toc59786027)

[3.2 演化态数据结构设计 11](#_Toc59786028)

[第四章 模块设计 12](#_Toc59786029)

[4.1 软件结构体系 12](#_Toc59786030)

[4.2 云化虚拟化网络演化对象建模模块 14](#_Toc59786031)

[4.3 云化虚拟化网络演化条件生成模块 14](#_Toc59786032)

[4.4 云化虚拟化网络演化规则分析模块 17](#_Toc59786033)

[4.5 云化虚拟化网络业务可靠度计算模块 24](#_Toc59786034)

[4.6 输入输出模块 25](#_Toc59786035)

[第五章 系统出错处理设计 28](#_Toc59786036)

[5.1 出错信息 28](#_Toc59786037)

[5.2 补救措施 28](#_Toc59786038)

[5.3 系统维护设计 28](#_Toc59786039)

# 引言

## 编写目的

为了更好的进行软件开发，必须严格地遵循软件工程规范，概要设计是其中不可缺少的一步。该设计说明书不仅是对数据结构、软件结构、处理过程设计方案的有效记载，而且还是后期的详细设计和编码的重要依据。

本报告编写是为了说明云化虚拟化网络可靠性评估方法的软件实现步骤，为下一阶段的软件详细设计、编码、测试提供参考依据；同时对编码、命名进行规范，统一软件各部分的输入输出接口。

本文档适用于所有与“基于网络演化模型的云化虚拟化网络可靠性评估”相关的人员，包括：北航网络可靠性课题组管理人员、项目评审组、项目经理、技术开发人员（包括分析人员、设计人员、程序编写人员）、测试人员。有关人员依据其职责应重点阅读本文档各部分或选择性阅读本文档。

## 背景



图 1 云化虚拟化网络示意图

软件系统名称：基于网络演化模型的云化虚拟化网络可靠性评估软件

云化虚拟化网络（如图 1所示）中采用了网络功能虚拟化 (NFV) 技术使得传统网络的单一软硬件系统变为基于通用硬件、分层解耦、多个网元共享的软硬件平台，网络系统的故障类型与故障发生后的动态变化更加复杂。总的来说，云化虚拟化网络的呈现三个特性：1.复杂异质性：网络不同层次的节点类型不一，不同节点存在着不同的故障与恢复行为；2.动态性：由于云化网络的中的VNF采用的不同的冗余及保护策略，因此VNF部署的实际网络节点会因为网络状态的改变而发生动态变化，这将导致业务路径的进一步动态变化；3.耦合性：不同的VNF因为部署的硬件设施发生耦合，同时不同的业务可能会调用同一VNF，这导致网络的业务进一步交联耦合。综上，以上的三个特性导致云化虚拟化网络的业务可靠性评估出现了难以建模分析的问题。

而通过调研我们发现，在现有对云化虚拟化网络的可靠性评估中主要采用了基于Petri网的衍生模型或者基于级联失效的模型。这些模型主要评估了抽象化的虚拟网络功能 (VNF) 故障后对网络可靠性的影响，不能对从硬件故障发生影响到顶层网络业务的角度进行综合分析；同时对故障发生后动态变化建模也只是针对VNF之间的相互影响，不能描述不同层级的故障同时发生后网络的动态变化行为[6-9]。在建模本身存在着难以描述云化虚拟化网络的状态空间爆炸问题。而现有的针对其他网络的仿真软件如OPNET/NS3等，并不专门针对可靠性进行仿真。故而如何针对不同SLA的业务，评估其在云化虚拟化网络中运行的可靠性以给与用户确信的保障，成为目前网络设备及运行方所需回答的一个重要问题。

因此，本软件依据课题组提出的网络演化模型，设计一个能够全面支持云化虚拟化网络的业务可用度计算工具。该软件针对影响云化网络动态演化的因素进行考虑，能够建立起云化虚拟化网络的模型，对影响网络业务的各种演化条件进行仿真，并计算出部署在云化虚拟化网络的各个业务以及整网业务的可用度，解决现存的计算工具和分析软件针对性不够、难以分析业务可用性的问题，对于当前网络可靠性研究、网络可靠度计算具有非常大的工程应用价值。软件在设计时采用模块设计，后期可以依据实际需求进行扩展。

## 定义

**NFV:** Network Functions Virtualization，网络功能虚拟化，是一种网络架构方式。

**VNF:** Virtualized Network Function，虚拟网络功能，是NFV中虚拟化的网元[1]。

**DCGW** Data Center Gateway，数据中心网关，作用相当于路由器[2]。

**EOR:** End of Row，服务器机柜内部交换机。

**TOR:** Top of Rack，服务器机柜的最上面安装接入交换机。

**SERVER:** 服务器，上面可以承载多个VNF。

**NCE：**Network Cloud Engine，云化网络引擎，主要负责对VM的管理与控制。

**NS:** Network service，网络服务，由多个VNF组合形成。

**SFC:** Service Function Chain，服务功能链，由不同NS进行连接形成。

**MTBF:** Mean Time Between Failure，平均无故障工作时间[3]。

**MTTR:** Mean time to repair，平均修复时间，由故障转为工作时修理时间的平均值。

**服务：**网络具有的某种能力，能够对外提供使用[4]。在本报告中对应NS。

**业务：**通过对服务的组合而达到对外提供使用的某种综合能力[4]。在本报告中对应SFC。

**过程性故障：**网络服务部署、业务配置、使用策略或使用方式等不合理造成的，不能落是在某个具体对象上的故障[4]。在本报告中主要指网络业务流量中断。

**业务可靠性：**以过程性故障为核心的网络支持能力[4]。这里规定条件是外部的演化条件，规定时间是单次仿真评估的时长，规定功能是网络业务保持连通的能力。

**整网业务可靠性：**部署在网络中的所有业务在规定条件下和规定时间内，完成规定业务功能的能力。

**业务可靠度**：用来横梁业务可靠性的指标，可由下式定义[5]：

业务可靠度=业务可用时间/(业务可用时间+业务恢复时间)

## 参考资料

[1] Lac C, Lac C, Adams R, et al. ETSI GR NFV-REL 007: Network Function Virtualisation (NFV);

Reliability; Report on the resilience of NFV-MANO critical capabilities[R]., 2017.

[2] Huawei. Configuration Guide VXLAN-NFVI Distributed Gateway[Z]. 2020.

[3] IEC. IEC 61907: Communication network dependability engineering[S]. 2009.

[4] 黄宁. 网络可靠性及评估技术[M]. 国防工业出版社, 2020.

[5] Schöller M, Khan N, Adams R, et al. ETSI GS NFV-REL 001: Network Functions Virtualisation (NFV); Resiliency Requirements[R]., 2015.

# 总体设计

## 需求规定

该软件对给定的云化虚拟化网络进行仿真，计算响应的业务可用度，并未后续的业务可用度优化设计提供支持。所需要的网络信息包括：云化网络的节点的数目与类型，节点上部署的VNF，网络的拓扑结构，各条链路的信息(包括源节点与宿节点)，故障模式信息(节点失效模式，节点失效的MTBF，故障检测率，故障检测时间，自动修复率，自动修复时间，人工修复的MTTR，迁移与倒换概率，迁移与倒换时间)，VNF在网络上的部署信息，以及业务(SFC)在网络上的部署信息等参数。这些信息以一个.xlsx文件进行输入。该软件最后会输出一个在给定时间周期下（如T = 20年）的每个网络业务的可用度值以及整网络业务可用度值，结果保存在一个.xlsx文件中。

## 运行环境

Windows 10, Linux和Mac OS操作系统

## 开发环境

### 硬件环境

建议硬件配置：Intel Core i7 CPU、16G RAM、40G ROM

### 软件环境

Python >=3.8.3

Networkx >= 2.5

Numpy >= 1.18.5

Pandas >= 1.1.4

Matplotlib >= 3.2.2

Microsoft excel（2003及以上版本）

### 开发语言

Python

## 基本设计概念

由于本软件最终所需评估的对象是网络中所有业务的可用度，因此我们在这对单业务可用度与整网业务可用度进行定义。

### 单业务可用度

参考《基于网络演化模型的云化虚拟化网络可靠性评估方法》报告，给出云化虚拟化网络业务*A*可用度的定义：



式中，为业务*A*在第*i*次仿真中在时间区间*[0,T]*的可用度值,为单次仿真内的单次业务故障时间，*T*为单次仿真时长。的业务故障需要根据关注的业务不同定义业务故障判据。

业务中断时间定义如下：



其中，为业务流量的归一化值，为业务流量的故障判据。

对于每个业务，由于部分输入的演化条件影响是随机分布的，因而需要做*N*次仿真，用仿真均值去估计业务的可用度值。此时，业务*A*可用度为：



### 整网业务可用度

对于整网业务可用度，定义一个评估整网可用度函数来定义整网业务可用度：



其中，为整网可用度，为业务的数目，为第*i*个业务的重要度值。如果相等的话，则整网业务可用度为每个业务可用度的均值。

## 功能结构设计

本软件的功能如图 3所示。主要实现以下功能：

1.数据读入功能：读入云化虚拟化网络数据文件（.xlsx文件）。

2.网络模型构建功能：建立在无故障发生事的网络模型

3.网络演化态生成功能：根据节点的相关故障信息，生成在网络状态发生改变的时间点下，网络触发网络演化的故障与修复节点。

4.网络状态演化功能：根据输入的网络演化态和网络模型，对网络状态进行演化，并计算每个演化态下的业务不可用时间。

5.业务可用度计算功能：计算在同时部署多个业务时，每个单业务的可用度值与整网业务可用度值。

6.图表化数据输出功能：以表格的形式输出单业务与整网业务可用度值，并输出演化后的网络拓扑与业务路径。



图 2 软件功能结构设计

# 数据结构设计

## 网络演化对象数据结构设计

对于云化虚拟化网络来说，其网络演化对象数据G主要基于networkx库中的networkx.Graph类进行实现的。该类包括了节点信息node\_attr\_dict\_factory和链路信息edge\_attr\_dict\_factory以及网络属性信息graph\_attr\_dict\_factory等。这些信息均是dict字典类型的数据结构。在此基础上，我们设定了网络节点信息node\_info，网络业务信息edge\_info和网络业务信息(包括VNF信息VNF\_info，服务信息Service\_info和业务信息Application\_info)。

网络节点信息是指网络中由DCGW、EOR、TOR、server等硬件设备或虚拟出来的VM等软件单元构成的网络基础组成成分。由于在云化虚拟化网络中，各个网络节点的节点类型相互异质，因此需要对不同的网络节点类型进行表述。在此，我们将网络节点类型划分为5类：DCGW节点、EOR节点、TOR节点、server节点和VM节点。同时用节点的状态表征节点当前状态是否故障。节点信息的示例如所示，红色部分为表格内容的示例。

表格 1 云化虚拟化网络Eode\_info的数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量类型 | 可空 | 描述 | 示例 |
| NodeID | 字符串 | 否 | 描述节点唯一标识符 | S1 |
| NodeType | 枚举型 | 否 | 描述节点的类型, 取值为(DCGW/TOR/EOR/Server/VM/Proc) | server |
| NodeFailType | 字符串 | 否 | 描述节点的故障模式类型 | SF1 |
| NodeFailMTBF | 浮点型 | 否 | 描述该故障类型的平均故障时间(h) | 43800 |
| NodeFailFDR | 浮点型 | 否 | 故障发生后的检测率 | 1 |
| NodeFailFDT | 浮点型 | 否 | 故障检测时间(h) | 0.004167 |
| NodeFailAFRR | 浮点型 | 是 | 对于软件类故障的自动维修概率，对于硬件节点是空值。 | 无 |
| NodeFailAFRT | 浮点型 | 是 | 对于软件类故障的自动维修时间(h)，对于硬件节点是空值。 | 无 |
| NodeFailMTTR | 浮点型 | 否 | 故障被检测到以后无法自动维修下，人工平均修复时间(h) | 2 |
| NodeFailMR | 浮点型 | 是 | 主备和Nway业务将其从故障节点迁移到新节点上的概率，server节点的属性，其余节点为空。 | 0.9 |
| NodeFailMT | 浮点型 | 是 | 主备和Nway业务将其从故障节点迁移到新节点上的时间(h)，server节点的属性，其余节点为空。 | 0.166667 |
| NodeMP | 字符串列表 | 是 | server节点的迁移控制链路（server到NCE的路），server节点的属性，其余节点为空。 | [S1，T1，E1，D1，D2，E3，T6，S6] |
| NodeState | 枚举型 | 否 | 表示节点是否出现故障。取值为0或1。1表示节点正常工作，0表示节点故障 | 0 |
| NodeIdle | 枚举型 | 否 | 表示节点是否空闲（未部署VM或业务）。取值为0或1。0表示节点空闲，1表示节点上部署了业务或VM | 1 |
| Tasp | 常量 | 否 | 通知巡检人员时间(h) | 0.5 |
| Tchk | 常量 | 否 | 巡检时间(h) | 168 |

网络链路信息是指网络中不同节点之间的流量通路。这里包含两种链路，一种是上层节点向下层节点发送数据的南北向链路，另一种链路是同种节点时间的相互控制、数据转移的东西向链路（例如对于Nway业务来说，其业务部署的VM节点要和流量均衡器的VM节点之间相互交换信息）。对于每种链路，均需要其源宿节点的信息。同时需要其链路容量与流量的信息。链路信息的数据字典示意如表格2所示，红色部分为表格内容的示例。

表格 2 云化虚拟化网络Edge\_info的数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量类型 | 可否为空 | 描述 | 示例 |
| EdgeID | 字符串 | 否 | 描述链路唯一标识符 | Eg1 |
| EdgeSourceNode | 字符串 | 否 | 链路的源节点 | S1 |
| EdgeDestinationNode | 字符串 | 否 | 链路的宿节点 | V1 |
| EdgeCapacity | 浮点型 | 否 | 链路容量（GBps） | 10 |
| EdgeTraffic | 浮点型 | 否 | 链路流量（GBps） | 8 |

表格 3 云化虚拟化网络业务信息Application\_info的数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量类型 | 可空 | 描述 | 示例 |
| ApplicationID | 字符串 | 否 | 描述业务的唯一标识符 | A1 |
| ApplicaitonServices | 字符串列表 | 否 | 描述业务所调用的服务功能链 | [Service1, Service2, Service3] |
| ApplicationWorkPath | 字符串列表 | 否 | 描述目前业务链路 | [D1，E1，T1，S1，V1，V2, V1，S1，T1，S2，V2,…] |
| ApplicationAvail | 浮点型 | 否 | 描述当前业务的可用度值 | 1.0 |
| ApplicationUnavailTime | 浮点型  字典 | 否 | 业务累计不可用时间 | 0 |
| ApplicationStatus | 布尔型 | 否 | 描述当前业务是否正常 | 1 |
| ApplicationInitTraffic | 浮点型 | 否 | 记录业务初始流量（Gbps） | 3.5 |
| ApplicationTraffic | 浮点型 | 否 | 描述当前业务流量相对值（比例） | 0.88 |
| ApplicationThresh | 浮点型 | 否 | 描述判断业务不可用的最低流量（比例） | 0.60 |

根据提供的信息，不同的VNF部署在如表格4所示，红色部分为VNF不同属性的实例。

表格 4 Service\_info数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量类型 | 可空 | 描述 | 示例 |
| ServiceID | 字符串 | 否 | 描述网元的唯一标识符 | Service1 |
| ServiceVNF | 字符串列表 | 否 | 描述VNF所调用的Service | [VNF1, VNF2, VNF3] |

每个的VNF部署在不同的节点上，不同的VNF部署如表格5所示。

表格 5 VNF\_info数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量类型 | 可空 | 描述 | 示例 |
| VNFID | 字符串 | 否 | 描述网元的唯一标识符 | VNF1 |
| VNFBackupType | 枚举型（主机型、主备型、Nway型） | 否 | 根据业务流量特征划分的三种Service备用路径类型 | 主备型 |
| VNFDeployNode | 字符串列表 | 否 | 当前业务运行时部署的主要进行节点 | V3 |
| VNFBackupNode | 字符串列表 | 否 | 当前业务所需的备用节点 | V7 |
| VNFFailSR | 浮点型 | 是 | 主备和Nway业务将其从故障节点倒换到新节点上的概率。当该节点无备份时为空值。 | 0.9 |
| VNFFailST | 浮点型 | 是 | 主备和Nway业务将其从故障节点倒换到新节点上的时间(h)，当该节点无备份时为空值。 | 0.166667 |
| VNFSwitchPath | 字符串列表 | 是 | 描述VNF控制链路（倒换）, VNF的属性，其余节点为空。 | [V3，S1，T1，E1，D1，D2，E3，T6，S6]；  [V3，S1，T1，E1，D1，E2，T3，S3，V5] |
| VNFWait | 布尔型 | 是 | 描述VNF的迁移或倒换是否处于等待,0为不等待，1为正在等待 | 1 |

## 演化态数据结构设计

以数据字典的形式对演化态数据evol进行说明，具体如下：

表格 6 网络演化态数据evol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | 变量类型 | 可空 | 描述 | 示例 |
| EvolTime | 浮点数 | 否 | 演化态时间 | 0.97 |
| EvolFailNodesSet | 字符串集合 | 是 | 当前演化态故障节点集合 | [‘V457’] |
| EvolRecoNodesSet | 字符串集合 | 是 | 当前演化态恢复节点集合 | [‘V457’] |

# 模块设计

## 软件结构体系



图 3 系统功能模块图

通过对本软件所要实现的系统功能进行分析，可以将系统的划分为以下5个模块：输入输出模块、云化虚拟化网络演化对象建模模块、云化虚拟化网络演化条件生成模块、云化虚拟化网络演化规则分析模块和云化虚拟化网络业务可靠性计算模块，如图 3所示。

整个系统的运行流程如图 4所示。



图 4 系统的运行流程图

## 云化虚拟化网络演化对象建模模块

功能：实现输入网络信息后，建立网络演化模型。

输入：网络的相关数据，包括节点数据node\_info、链路数据edge\_info、VNF数据VNF\_info、Service数据Service\_info和业务数据Application\_info.

输出：演化对象，包括初始状态下的网络模型G（节点信息、边信息）、业务服务信息（VNF信息、Service信息与Application信息），其中，业务服务信息属于软件应用信息。

流程如下：



图 5云化虚拟化网络演化对象建模流程

## 云化虚拟化网络演化条件生成模块

功能：实现对网络演化条件的生成。

输入：云化网络仿真模型。

输出：网络演化态数据evol

算法设计流程：其中，t\_r为故障修复时间，t\_f为故障前时间



图 6 云化虚拟化网络演化条件生成模块算法流程

## 云化虚拟化网络演化规则分析模块

**1.Server状态变化**：

当Server节点状态变化，先判断该节点是故障还是修复；若为修复，则遍历Server上所有的VM，认为该Server上所有VM修复，转入到VM修复的处理方式；若为故障，则同样遍历Server上所有的VM，认为该Server上所有的VM故障，转入到VM故障的处理方式；最后再判断控制链路是否故障，若控制链路故障，则Server无法迁移；若控制链路正常，则Server迁移。

单点Server状态发生变化以后，业务的不可用时间计算流程如下：



图 7 Server节点故障的故障演化规则

Server节点的迁移策略如下：判断控制链路是否故障，若控制链路故障，则先等待控制链路修复，再将该故障Server上的所有VM迁移到资源池中的空闲Server上，更新当前网络状态。

单点Server故障以后，业务的不可用时间计算流程如下：



图 11 Server节点迁移策略

1. **Vswitch状态变化**

当Vswitch节点状态发生变化以后，先判断是该Vswitch节点是故障还是修复。若为修复，则遍历其控制的所有VM，认为其控制的所有VM修复，转入VM节点的修复处理；若为故障，则同样遍历其控制的所有VM，认为其控制的所有VM故障，转入VM故障的处理方式。



图 8 Vswitch节点故障

1. **VM状态变化**：

当VM节点状态发生变化以后，先判断该VM节点是故障还是修复？如果节点是修复，则将当前等待该节点修复的业务状态变为正常，记录业务中断的时间（见VM节点的修复处理，接口①）；如果该VM节点是故障，则遍历该节点上的所有业务，如果为主机型业务，将该业务状态变为故障，记录该业务故障开始时间（见主机型业务的故障处理，接口②）；如果为主备型业务，则先判断控制链路是否故障？若控制链路故障，则将该业务状态变为故障，记录该业务故障开始时间，等待控制链路修复；若控制链路正常，则业务倒换到备用VM上，同时备用VM升主、故障VM变备，记录业务故障时间，等待故障修复（见主备型业务的故障处理，接口④）；如果其上承载为Nway型业务，则判断Nway型业务的所有链路是否都中断？若都中断，则该业务故障，记录故障时间；若没有都中断，则该业务正常（见N-way型业务的故障处理，接口⑤）。

VM故障以后，业务的不可用时间计算流程如下：











图 9 VM节点状态变化

1. **交换机状态变化**

图 10 EOR/TOR/DCGW节点故障后的演化规则

此模块应用于NodeType = EOR/TOR/DCGW。根据当前NodeID查询到该节点对应的节点状态NodeState，当节点状态由1变为0时，意味着该节点故障，此时记录当前演化时间T0，并遍历所有业务。如果一项业务的工作路径含有该节点，则将此时业务流量记录后，将该业务流量归零，并回到等待NodeState变化。如果一项业务下任何一VNF的控制路径含有该节点，并其中有节点在进行倒换，则转移至下层VM故障处理模块，如没有节点在进行倒换，继续遍历业务，直至所有业务遍历完成。

当检测到该节点NodeState由0变为1时，则以为该节点恢复，此时记录当前时间为T1，如该业务路径是否存在其它节点故障，则转移至VM故障处理，如果不存在其它节点故障的话，此时恢复业务流量至LastNormalTraffic，记录业务中断事件并将其加至业务中断时间，完成后继续遍历业务，直至所有业务遍历完成。

## 云化虚拟化网络业务可靠度计算模块

功能：实现对给定时间和计算次数的单业务与整网业务可用度计算。

输入:计算次数N、计算周期T、网络初始模型G(0)

输出:单业务可用度与整网业务可用度值ApplicationAvail\_whole

算法设计：



图 12云化虚拟化网络业务可靠度计算模块算法

## 输入输出模块

功能：用户通过给定云化网络的输入，对云化网络的参数进行赋值，软件通过模块化的计算，输出得到在给定时间周期下的每个网络业务的可用度值以及整网络业务可用度值，并呈现不同时间下网络业务可靠度的变化趋势图。

输入：云化网络的节点的数目与类型，节点上部署的VNF，网络的拓扑结构，各条链路的信息(包括源节点与宿节点)，故障模式信息(节点失效模式，节点失效的MTBF，故障检测率，故障检测时间，自动修复率，自动修复时间，人工修复的MTTR，迁移与倒换概率，迁移与倒换时间)，VNF在网络上的部署信息，以及业务(SFC)在网络上的部署信息等参数。

输出：给定时间周期下的每个网络业务的可用度值以及整网络业务可用度值，以及网络业务可靠度随时间变化的折线图。

输入输出界面：



图 13 云化虚拟化网络操作界面框架

框架：

Python 提供了多个图形开发界面的库，几个常用 Python GUI 库如下：

Tkinter： Tkinter 模块(Tk 接口)是 Python 的标准 Tk GUI 工具包的接口。Tk 和 Tkinter 可以在大多数的 Unix 平台下使用,同样可以应用在 Windows 和 Macintosh 系统里。Tk8.0 的后续版本可以实现本地窗口风格,并良好地运行在绝大多数平台中。

wxPython：wxPython 是一款开源软件，是 Python 语言的一套优秀的 GUI 图形库，允许 Python 程序员很方便的创建完整的、功能健全的 GUI 用户界面。

Jython：Jython 程序可以和 Java 无缝集成。除了一些标准模块，Jython 使用 Java 的模块。Jython 几乎拥有标准的Python 中不依赖于c语言的全部模块。比如，Jython 的用户界面将使用 Swing，AWT或者 SWT。Jython 可以被动态或静态地编译成 Java 字节码。

我们这里选择Tkinter进行界面开发，Tkinter是Python自带的GUI库，它最好的一个优点就是不用另外装额外的库拿来就可以用了，而且整个库很小巧，入门简单，适合于解决实际问题。

# 系统出错处理设计

## 出错信息

本节用一览表的方式说朗每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

|  |  |
| --- | --- |
| **出错或故障情况** | **系统输出信息的形式与、含意及处理方法** |
| 文件无法导入 | 系统弹出对话框，表明导入的文件数据与选择的可靠度计算类型或是输入的试验相关数据不相符合。 |
| 无法导出可靠度曲线 | 输入试验相关数据有误或是与导入的文件数据不相符合 |
| 按钮不可用 | 按钮不可用，说明操作流程还未到相应步骤，所以请按相应操作步骤执行 |
| 可靠度曲线生成错误 | 文件重新导入需要重新输入试验参数 |
| 软件无法得到输出 | 输入信息过大，计算过于复杂需要简化参数输入 |

## 补救措施

1. 操作错误：按照后期的用户手册进行正确操作；
2. 系统错误：在测试阶段进行功能测试，将每一步测试中的错误以表格形式记录，并记下相应的处理方法。使用阶段出现的功能错误，紧急情况下可重启软件，并将错误记录并上报开发人员。

## 系统维护设计

为了系统维护的方便而在程序内部设计中做出的一定安排，包括在程序中专门安排用于系统的检查与维护的检测点和专用模块。由于系统模块间的联系很松散，故在系统维护的时候可以很好的通过分模块测试与维护。