基于 Python 混合编程

薛建波，谭凌岚，罗佳
（武汉虹旭信息技术有限责任公司，湖北武汉，430074）

摘要：Python 不断的发展壮大，应用范围非常广泛，功能多样，有多种完整成熟的架构，适于短期开发。Python 是一种解释性语言，拥有完善的标准库和接口，可以进行混合编程。

关键词：Python；web 开发；混合编程
DOI:10.16520/j.cnki.1000-8519.2018.12.044

PYTHON BASED MIXED-LANGUAGE PROGRAMMING

Xue Jianbo,Tan Linglan,Luo Jia
(Wuhan Hong Xu Information Technology Co., Ltd, Wuhan Hubei, 430074)

ABSTUCTS: With the continuous development of python ,it is wildly used and is versatile ,also has full and mature Template .Which is suited for short-term development .Python is a kind of interpreted language and has complete standard library and interface ,which is the best choice of mixed-language programming.
Keyword:Python;Web development;Mixed-language programming

1 Python 语言简介

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。其设计具有很强的可读性，相比其他语言，Python 具有特有的语法结构。Python 支持广泛的应用程序开发，从简单的文字处理到 web 开发，以及数据库操作。Python 的特点有以下几个方面：Python 有相对比较少的关键字，结构简单，语法定义明确。Python 代码使用空格对齐，更加便于阅读和理解。每一个模块有清晰的定义和调用说明，在软件完成以后的维护很容易。Python 的易于学习和编写使其快速地流行。经过多年的发展和标准制定，Python 提供了丰富的标准库。这些库文件安装可以通过 Python 命令进行。库文件的调用也十分简单。如果标准库中没有目标程序源码，用户可以自己通过 c 或者 C++ 自我实现。也可以通过 API 对 C 和 C++ 进行调用。Python 支持所有的商业数据库的数据接口，支持 GUI 编程。提供成熟可靠的的 web 框架，简化的 web 开发的过程。不同用途或者功能的软件开发可能需要不同的语言进行编写。特定的语言提供了良好数据操作和简化后的编程逻辑。而 Python 提供了当前许多不同功能软件开发、操作的库。不同的程序，相同的语言进行实现，效率更高，兼容性更好。此外，Python 还支持机器学习和创建数学模型。

2 基于 Python 的混合语言编程

使用 Python 进行混合编程有两种方式：扩展和嵌入。扩展是通过 C、C++ 等系统语言实现 Python 的功能模块。嵌入是将 Python 解释器加载到应用程序中，使程序能够解释运行 Python 语言写成的脚本。两者都需要调用 C 语言应用程序接口。

2.1 需要扩展 Python 语言的理由

（1）添加 / 额外的（非 Python）功能，提供 Python 核心功能中没有提供的部分，比如创建新的数据类型或者将 Python 嵌入到其它已经存在的应用程序中，则必须进行扩展编译。（2）性能瓶颈的效率提升。解释型语言一般比编译型语言慢，如果某一个模块处理的数据比较多，或者会频繁的操作 IO，整个程序的运行速度会降低，形成程序的瓶颈。但是如果将所有的程序都用低级语言编写，会存在两个问题：一是工程量太大，不符合软件开发的初衷 - 高效率；二是有些模块，Python 的运行效率与低级语言运行效率相差不大，不需要重新编译。（3）核心代码加密。因为 Python 是解释性语言，源代码没有私密性。将核心代码由 Python 语言转变为编译语言就变得很重要。

2.2 扩展的典型程序结构

（1）创建应用程序的源代码。（2）利用样板来包装代码。（3）创建 setup.py 进行编译，封装为 Python 的库。（4）通过 Python 解释器运行。

3 应用举例

下面以在 Linux 下面环境下进行文本检索的实例，介绍基于 Python 的混合编程。该程序实现了对格式化的文本进行检索，查找关键字，提取出包含关键字的数据段。程序运行环境：操作系统 -Red Hat 4.4.7-3；编译器 -GCC 4.4.7；Python 解释器 -Python 2.6.6。

3.1 创建程序源代码

在头文件中添加 Python.h，里面包含了 Python 定义的所有的内部数据结构和 C API 函数原型。按照程序逻辑编写文本处理函数体。通过 GCC 编译其中的主程序。

3.2 进行模块封装

（1）对照 C 语言源代码，在文件中为每个模块的每一个函数增加 PyObject* M_func() 的包装函数。包装函数的目的是把 python 的值传递给 c，再把 c 中函数的计算结果转换成 Python 对象返回给 python。

```
static PyObject * Extest_func(PyObject *self, PyObject *args) {
  int res;//计算结果值
  int num;//参数
  PyObject* retval;//返回值
```

```
  //i表示需要传递进来的参数类型为整型，如果是，就赋值给num，如果不是，返回NULL；
  res = PyArg_ParseTuple(args, "i", &num);
  if (!res) {
    //包装函数返回NULL，就会在Python调用中产生一个TypeError的异常
    return NULL;
  }
  res = func(num);
  //需要把c中计算的结果转成python对象，i代表整数对象类型。
  retval = (PyObject *)Py_BuildValue("i", res);
  return retval;
}
```

（2）将每个模块增加一个 PyMethodDef ModuleMethods[] 的数组，为 Python 解释器提供调用入口：

```
static PyMethodDef
ExtestMethods[] = {
  {"func_M", Extest_m, METH_VARARGS},
  {"func_F", Extest_f, METH_VARARGS},
  {"func_S", Extest_s, METH_VARARGS},
  {NULL, NULL},
};
```

（3）增加模块初始化函数 void initMethod()：

```
void initExtest() {
  Py_InitModule("Extest", ExtestMethods);
}
```

3.3 编译测试

（1）编写 setup.py 文件。

```
#!/usr/bin/env python
from distutils.core import setup, Extension
  MOD = 'Extest'     程序扩展名                              源C程序
  setup(name=MOD, ext_modules=[Extension(MOD, sources=['Extest.c'])])
```

（2）执行 setup 文件，扩展的模块会被导入至 Python 的调用库。程序执行的时候，解释器找到模块的位置，进行调用。

（3）运行结果。在文本中找到的关键字数据，并且提取出了文本段。

在原始的文本数据量很大的情况下，扩展以后的数据检索速度明显快于纯 Python 脚本检索的速度。在此基础上，加入 Python 的机器学习的库文件进行程序功能的扩展，就能实现数据推荐功能。

4 结语

人工智能兴起带动了 Python 的发展。Google 开源了 Python 机器学习源码，提供大量标准的 Python 机器学习库。Python 的底层是由 C/C++ 实现。运行速度低于编译语言。随着处理的数据量越来越大，对程序处理速度要求也会不断提升。混合语言编程以后肯定会应用的更加广泛。

参考文献

[1]Kalle Lyytinen. Different Perspectives on Information Systems:Problems and Solutions.ACMComputing Surveys,Vol.19,No.1,March 1987.
[2]Larry Wall, Tom Christiansen. Programming Perl, Second Edition[M].O'Reilly and Associates,1996.
[3]John Ousterhout.Scripting:Higher Level Programming for the 21stCentury[J].IEEE ComputerMagazine,1998, 31(3):23～30.

（上接第 63 页）

2.4 信息发送的控制要点

电子信息在借助低速传输通道进行传输的时候常会出现信息阻塞问题，这时可以通过通信控制软件来将各类信息按照紧急度和重要性划分为不同的优先级，并且为其设置相应的独立缓冲序列，按照优先级的高低顺序逐步完成电子信息的提取。此外也可以通过控制流量的方式进行信息的发送，当新信息进入排队时，将会替换优先级比自己低的已丧失使用价值、缓冲区中排队过久的旧信息，有效地解决了不同带宽信道间信息拥塞问题。

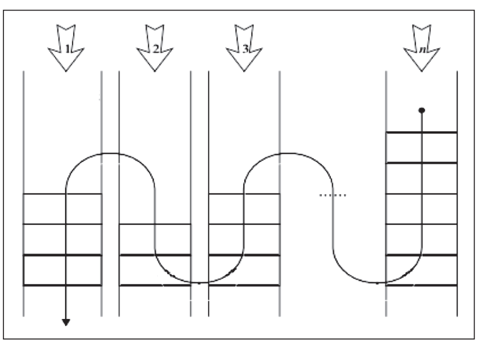

图 1　有序有效的信息发送流程图

2.5 虚拟网安全的控制要点

对于虚拟网的安全控制要点就在于结合所有用户信息资源的访问情况进行具体划分为若干个 VLAN，在每个 VLAN 中可以进行多播，同时工作可以通过绑定端口、设置防火墙、设置防火墙等多种方式对不同的 VLAN 进行控制，防止各环节之间发生信息的交互。应用 VLAN 能够全面保证信息的传输的安全，同时还可以进一步促进系统对网络信息资源整体控制水平的提高[2]。

3 结语

电子信息系统中信息传输技术的应用对于系统的安全运行，更好的满足用户要求等方面都具有重要的应用意义，因此相关工作人员还需要的日常工作的过程中不断发挥创新思维，加大实践力度，全面提高信息传输的控制水平。

参考文献

[1] 侯金香，侯海新．浅谈计算机电子信息系统中信息传输控制技术的应用 [J]. 通讯世界,2017(19):84-85.
[2] 阴骏．计算机电子信息系统中信息传输控制技术探索 [J]. 无线互联科技,2012(05):83.