

基于 Web 的 Python3 编程环境^①

刘志凯¹, 张太红^{1,2}, 刘磊¹, 罗鹏¹

¹(新疆农业大学 计算机与信息工程学院, 乌鲁木齐 830052)

²(中国农业大学 信息与电气工程学院, 北京 100083)

摘要: 为了简化编程环境, 增强编程体验, 提出了一种基于 web 的 python3 编程环境. 该应用的 web 环境采用基于 python 的 Django 框架, 通过将 python3 代码转换成浏览器可执行的 JavaScript 脚本, 达到在浏览器中运行 python3 代码的目的. 与传统的 python3 编程环境相比, 该编程环境不用在本地安装部署任何开发环境, 即可实现远程编程和代码共享. 实验结果表明, 该编程环境可行性强, 具有良好的稳定性和并发性.

关键词: python3; web; 编程环境; 远程代码共享; Django

Python3 Programming Environment Based on Web

LIU Zhi-Kai¹, ZHANG Tai-Hong^{1,2}, LIU Lei¹, LUO Peng¹

¹(College of Computer & Information Engineering, Xinjiang Agricultural University, Urumqi 830052, China)

²(College of information and Electrical Engineering, China Agricultural University, Beijing 100083, China)

Abstract: In order to simplify the programming environment, and enhance programming experience, we propose a web-based python3 programming environment. The web environment of this application uses python-based Django framework. This application can achieve the purpose of running python3 code in the browser by converting the python3 code into browser executable JavaScript script. Compared with traditional python3 programming environment, this programming environment can achieve remote programming and code sharing without installing and deploying any development environment locally. Experimental results shows that our programming environment has strong feasibility, good stability and good concurrency.

Key words: python3; web; programming environment; remote code-sharing; Django

开发环境是一个为计算机程序员进行软件开发提供所有必要功能的软件应用程序,它通常包括源代码编译器,构建自动化工具和调试器.随着云计算技术的发展,PaaS(平台即服务)越来越受到开发者的青睐.所谓 PaaS 实际上是指将软件研发的平台作为一种服务,并提供给用户.本文在此背景下提出基于 web 的 python3 编程环境,为用户提供快速开发 python3 所需的应用和产品.

文献[1]表明,在美国顶尖高校中,python 已经成为教授计算机科学入门课程方面最受欢迎的语言.由于主流 web 编程环境缺少对 python3 的支持,所以开发出基于 web 的 python3 编程环境显得尤为重要.

针对传统 python3 编程环境需要在本地安装的现状,开发设计了基于 web 的 python3 编程环境,将 python3 转换成 JavaScript,作为 web 浏览器的脚本语言^[2].对该编程环境进行了性能测试,测试结果表明,该编程环境在保证程序正确编译运行的同时,也具有很好的稳定性和反应速度.

本文第 1 节分析了近年来国内外的相关研究工作;第 2 节对该编程环境的系统设计进行了详细介绍;第 3 节介绍了实现该编程环境的关键技术;第 4 节给出测试结果和分析;最后在第 5 节对全文进行总结并提出下一步需要继续深入研究的工作.

^① 收稿时间:2014-10-29;收到修改稿时间:2015-01-03

1 相关工作

软件开发已经成为当今世界不可或缺的一项重要活动, 软件开发的编程环境也几经更迭——从最初的记事本到多功能文本编辑器, 从命令行到集成开发环境. 然而这种基于本地的开发环境已经不能满足人们日益增长的编程需求, 因此, 如何不依赖本地开发环境而又能方便快速的进行开发成为一个值得关注和研究的问题. 其中基于 web 的编程环境成为国内外学者研究的热点^[3-6].

2008 年 10 月 6 日, Remy Sharp 开发了 JS Bin, 即最早的支持实时更新 JavaScript, CSS 和 HTML 的环境, 这也成为最早的基于 web 的编程环境. 目前虽然基于 web 的编程环境得到了飞速发展, 并且出现了不错的商业产品, Cloud9、Compilr 和 Repl.it 都是国外通用的在线编译软件, 前两个适合有一定编程经验的开发者, 对于编程初学者来说操作过于复杂, 值得一提的是, 他们都只支持 python2. Compileonline 虽然支持 python3, 但它不提供在线保存和分享功能. 与国外相比, 国内对基于 web 的编程环境研究较少, 缺乏相关技术文档和成熟产品, 本文提出的基于 web 的 python3 编程环境在准确性和稳定性方面都有不错的表现, 同时也弥补了国内相关研究的空白, 为后来的研究人员提供了参考.

2 系统设计

本系统设计的主要目的是为编程人员提供方便、高效的 python3 编程环境, 编程人员只需通过浏览器即可进入 python3 编程环境界面, 系统总体架构如图 1 所示.

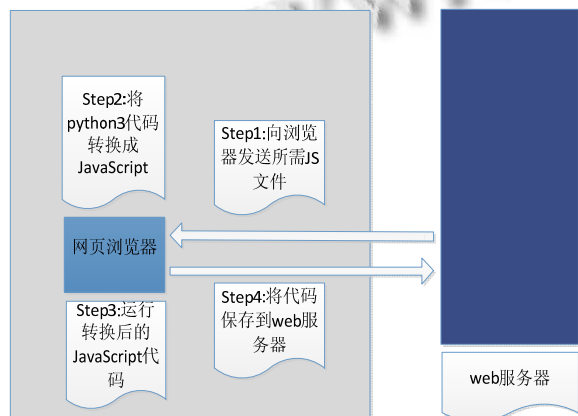


图 1 基于 web 的 python3 编程环境架构

2.1 系统架构

基于 web 的 python3 编程环境执行步骤如下:

步骤 1: 浏览器从服务器上读取所需 JS 文件.

步骤 2: 在浏览器中将 python3 代码转换成 JavaScript 脚本代码.

步骤 3: 在浏览器中运行转换后的 JavaScript 脚本.

步骤 4: 将代码保存到 web 服务器上.

2.2 技术实现

本系统在设计实现上主要采用了以下几种技术:

1) Django: 本系统底层采用开源的 web 应用框架—Django, 它包括一个模式系统, 对象相关的映射和用于动态创建管理界面的结构, 而且该框架是由 python 编写的, 所以有关 python 的接口和 API 文档都非常完善^[7].

2) Python Script: 众所周知 JavaScript 是网站开发中常用的脚本语言, 本系统尝试了一种新的方法, 将 python3 转换成 JavaScript, 作为 web 浏览器的脚本语言, 关键技术实现将在第 4 小节中详细介绍.

3) CodeMirror: 它是一个用 JavaScript 实现的代码编辑器组件, 专门用来嵌入到 web 页面以实现程序代码的高亮显示^[8]. CodeMirror 提供了丰富的 API 文档和 CSS 样式主题, 可以满足所有主程序语言的需求. 在本系统中采用 python3 的样式, 达到了很好的用户体验效果.

4) jQuery: 它是为简化 HTML 的客户端脚本而设计开发的一个跨平台的 JavaScript 库. jQuery 的语法设计, 使其能更方便地浏览文件, 选择 DOM 元素, 创建动画, 处理事件, 并开发 Ajax 应用程序. 本系统中采用多个 jQuery 库, 加快了系统的运行速度.

3 关键技术

从系统流程图可以看出, 本编程环境的编译过程在客户端的浏览器中完成, 需要保存时才涉及到 web 服务器, 具体实现主要有以下三个方面.

3.1 python3 转换成 JavaScript

所谓的基于 web 的 python3 编程环境, 具体指的就是在浏览器中编译执行 python3 代码. 我们知道浏览器最擅长执行的语言就是 HTML 和 JavaScript, 本编程环境就是将 python3 代码转换成 JavaScript, 然后再对其进行编译, 转换过程示例代码如下:

```
function run_Python(editor,result)
```

```

{
    .....
    document.getElementById(result).value = "";
    var val = "import sys\n";
    val += "from browser import document,prompt\n";
    val += "def raw_input(data):\n";
    val += "    return prompt(data)\n";
    val += "def write(data):\n";
    val += "document['"+result+"'].value += str(data)";
    val += "\n"+"sys.stdout.write = write";
    val += "\n" + "sys.stderr.write = write";
    val += "\n" + editor.getValue();
    var py_script= document.createElement('script');
    py_script.setAttribute("id","tempPython");
    py_script.style.visibility = 'hidden';
    py_script.type = 'text/Python3';
    py_script.textContent = val;
    document.body.appendChild(py_script)
    .....
}

```

3.2 运行 python3 代码

将 python3 代码转换成 JavaScript 后, 需要对其进行编译运行, 本文引入 brython.js, 实现了在浏览器中运行 python3 代码, 下面是一个在 HTML 页面中通过加载 brython.js 来运行 python3 代码的例子:

```

<html>
  <head>
    <script src="brython.js"></script>
  </head>
  <body onload="brython()">
    <script type="text/Python">
      from browser import document, alert
      def echo(event):
          alert(document["zone"].value)
          document['mybutton'].bind('click', echo)
    </script>
  <input id="zone">
  <button id="mybutton">click !</button>
  </body>
</html>

```

3.3 远程代码分享

将编程环境与 web 相结合, 一方面可以借助 web 平台进行快速开发, 另一方面就是实现了远程代码分享.

借助 Django 应用框架, 当用户保存时通过 uuid 来生成包含用户代码的唯一页面, 用户可以将该页面的链接发给其他人, 从而实现远程代码分享, 保存示例代码如下:

```

def save(request):
    ss = request.REQUEST.get('code_area,')
    id = str(uuid.uuid1())
    outfile = open('/home/.../data/'+id+'.py','w')
    outfile.write(ss)
    outfile.close()
    return HttpResponseRedirect('.....'+id)

```

其中, uuid(Universally Unique Identifier)是 128 位的全局唯一标识符, 通常由 32 字节的字符串表示. 本示例中使用的是基于时间戳的 uuid1(), 它是由 MAC 地址、当前时间戳、随机数生成, 可以保证全球范围内的唯一性.

4 系统测试

为检测该编程环境的系统性能, 采用 ApacheBench(Apache 旗下的一个测试工具, 简称 ab) 对该系统进行测试. 该测试工具可以同时模拟多个并发请求来对服务器进行压力测试, 用命令执行示例如下:

```
ab -n 100 -c 10 http://localhost:8000/index.html/
```

其中 -n 100 代表总的请求数, -c 10 代表每次发送的请求数, 上面这个命令的意思就是启动 ab, 向 localhost:8000/index.html/ 发送 100 个请求, 并每次发送 10 个请求——也就是说总共要发送 10 次.

图 5 表示的是向本地服务器发送 100 次请求, 且每次发送 10 个请求时, 用户所需的访问时间, 其中横坐标表示并发访问的请求次数(单位: 次), 纵坐标表示并发访问所需要的时间(单位: 毫秒).

由图 2 可以看出, 随着访问请求次数增加, 访问所需时间也在递增, 但所需时间都在 1 秒左右, 具有较高的稳定性, 这是因为本编程环境的编译和执行过程都在用户的浏览器端, 只有在保存时才会与服务器建立关联, 所以服务器端可保持较高的稳定性和并发

性.

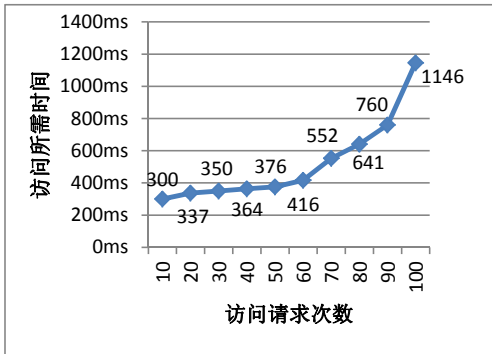


图 2 并发访问测试

5 总结

对于那些从来没有学习过编程或者并非计算机专业的编程学习者而言, python 已经成为最好的选择之一。随着互联网的飞速发展, 全新的在线教育平台 MOOC(Massive Open Online Courses)也被搬到了台前, 而其中最著名的三个提供商 edX, Coursera 和 Udacity 也全都提供了 python 语言的入门课程^[9]。本文设计并开发出了一种基于 web 的 python3 编程环境, 一方面可以让编程初学者把更多的精力放在编程本身上, 免去本地安装和配置编程环境的过程; 另一方面, 本系统还可作为在线教育平台的教学工具, 可以提供代码的编写、测试和远程共享功能。未来的工作重点是改进和完善系统功能, 以满足更广泛的用户需求。

参考文献

- 1 Philip G. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. <http://cacm.acm.org/blogs/blog-cacm/176450-Python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>. [2014-07-07].
- 2 Brython. <http://www.brython.info/doc/en/index.html/>. [2014-04].
- 3 Dutta M, Sethi KK, Khatri A. Web based integrated development environment. International Journal of Innovative Technology and Exploring Engineering, 2014, (3).
- 4 Kats LCL, Vogelij R, Kalleberg KT, et al. Software development environments on the Web: a research agenda. TUD-SERG, 2012, (014): 99-116.
- 5 Harshal S, Ajay RCA, Sharad R. Compiler as service over cloud. International Journal of Computer Applications, 2013, 70(1).
- 6 Arshad AM, Arshiya K, Sana S, Zainab M. Compilers on cloud. International Journal of Engineering Research & Technology (IJERT), 2013, 2(9).
- 7 Grehan R. Pillars of Python: Six Python Web frameworks compared. InfoWorld.com, 2011, (8).
- 8 CodeMirror. <http://codemirror.net/>. [2014-04].
- 9 郝丹. 国内 MOOC 研究现状的文献分析. 中国远程教育, 2013,(11):42-50.