



第14章 其他内核的配置

Jupyter金融应用从入门到实践

在Anaconda图形化界面中，R包含在Anaconda导航器中的RStudio选项中，单击“安装”（Install）按钮即可安装。如果计算机上没有图形化界面，如Linux服务器版，仍然可以用命令行的方式来安装R内核。

```
1.    $ conda install r-essentials  
2.    $ conda update r-essentials
```

- 使用命令行方式下载，则在命令行中运行以下命令。

```
$ wget -c
```

```
https://julialang-s3.julialang.org/bin/linux/x64/1.5/julia-1.5.3-linux-x86_64.tar.gz
```

- 通过上述两种方式下载好安装包后，进入下载好的Julia安装包所在位置。注意下载好的安装包是压缩包格式，要先进行解压。

```
$ tar xzvf julia-1.5.3-linux-x86_64.tar.gz
```

- 配置Julia。

将解压后的文件夹“julia-1.5.3”复制到/opt目录下。

```
$ sudo cp -r julia-1.5.3 /opt/
```

在/usr/local/bin中建立Julia的链接。

```
$ sudo ln -s /opt/julia-1.5.3/bin/julia /usr/local/bin/julia
```

- 检查是否安装成功。

在终端中查看Julia的安装版本信息，如图14-8所示。

```
$ julia -version
```


在安装好Node.js和npm后，接下来安装Jupyter对JavaScript的支持。

在Ubuntu终端窗口中输入以下命令，结果如图所示。

```
$ npm i -g ijavascript
```

```
(base) jupyter@ubuntu:~$ npm i -g ijavascript
/home/jupyter/anaconda3/bin/ijjs -> /home/jupyter/anaconda3/lib/node_modules/ijavascript/bin/ijavascript.js
/home/jupyter/anaconda3/bin/ijjsconsole -> /home/jupyter/anaconda3/lib/node_modules/ijavascript/bin/ijjsconsole.js
/home/jupyter/anaconda3/bin/ijjsinstall -> /home/jupyter/anaconda3/lib/node_modules/ijavascript/bin/ijjsinstall.js
/home/jupyter/anaconda3/bin/ijjskernel -> /home/jupyter/anaconda3/lib/node_modules/ijavascript/lib/kernel.js
/home/jupyter/anaconda3/bin/ijjsnotebook -> /home/jupyter/anaconda3/lib/node_modules/ijavascript/bin/ijjsnotebook.js

> zeromq@5.2.0 install /home/jupyter/anaconda3/lib/node_modules/ijavascript/node_modules/zeromq
> node scripts/prebuild-install.js || (node scripts/preinstall.js && node-gyp rebuild)

/home/jupyter/anaconda3/lib
├─┬─ ijavascript@5.2.0
│   ├── jp-kernel@2.0.0
│   ├── jmp@2.0.0
│   ├── zeromq@5.2.0
│   ├── nan@2.14.2
│   ├── prebuild-install@5.3.6
│   ├── detect-libc@1.0.3
│   ├── expand-template@2.0.3
│   ├── github-from-package@0.0.0
│   ├── minimist@1.2.5
│   ├── mkdirp-classic@0.5.3
│   ├── napi-build-utils@1.0.2
│   ├── node-abi@2.19.3
│   ├── semver@5.7.1
│   ├── noop-logger@0.1.1
│   ├── npmlog@4.1.2
│   ├── are-we-there-yet@1.1.5
│   ├── delegates@1.0.0
│   ├── readable-stream@2.3.7
│   ├── core-util-is@1.0.2
│   ├── isarray@1.0.0
│   ├── process-nextick-args@2.0.1
│   ├── string_decoder@1.1.1
│   ├── util-deprecate@1.0.2
│   ├── console-control-strings@1.1.0
│   └─┬─ gauge@2.7.4
│       ├── lodash@4.17.15
│       ├── ansi-regex@2.0.0
│       ├── ansi-styles@3.2.1
│       ├── chalk@2.4.1
│       ├── cli-cursor@2.1.0
│       ├── cli-spinners@2.1.0
│       ├── strip-ansi@4.0.0
│       └── strip-ansi@5.2.0
```

- 用以下命令下载 Scala

```
$ wget https://downloads.lightbend.com/scala/2.13.4/scala-2.13.4.tgz
```

- 安装 Scala

```
$ tar -xvf scala-2.13.4.tgz -C /usr/local/
```

- 配置 Scala

```
# Scala
export SCALA_HOME=/usr/local/scala-2.13.4
export PATH=$PATH:$SCALA_HOME/bin
```

图 14-23 ~/.bashrc 文件

编辑.bashrc 文件并在文件末尾添加 Scala 信息，

如图 14-23 所示。

```
$ vim ~/.bashrc
export SCALA_HOME=/usr/local/scala-2.13.4
export PATH=$PATH:$SCALA_HOME/bin
```

使用 source 命令使配置生效。

```
$ source ~/.bashrc
```

- 检测 Scala 是否安装成功

```
$ scala
```

Spark是专为大规模数据处理而设计的快速通用的计算内核，其主要目的是处理实时生成的数据。Spark是类Hadoop MapReduce的通用并行框架，Spark拥有Hadoop MapReduce的优点。但不同于MapReduce的是，在Spark中，Job中间输出的结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的场景。

安装过程较复杂，详细步骤见书本第十四章。



谢谢！