

如何让Scala脚本快速运行

本博客前两篇文章介绍了如何在脚本中使用Scala（[《在脚本中运行Scala》](#)、[《在脚本中使用Scala的高级特性》](#)），我们可以在脚本里面使用Scala强大的语法，但细心的同学可能会发现每次运行脚本的时候会花上一大部分时间，然后才会有结果。我们来测试下面简单的Scala脚本：

```
#!/bin/sh
exec scala "$0" "$@"
!#

println("Hello, iteblog!!")
args foreach println
```

我们将这段代码保存到test.sh文件里面，然后运行它：

```
[iteblog@www.iteblog.com iteblog]$ time sh test.sh
Hello, iteblog!!

real 0m1.070s
user 0m0.380s
sys 0m0.031s
```

我们可以看出，这么简单的程序都花费了1.07s才输出结果。这是因为每次运行的时候都会编译里面的内容，编译完才会运行脚本。值得高兴的是，我们可以将编译的内容保存下来，然后以后运行的速度就会加快。Scala解析器提供了`-savecompiled`参数，可以将编译的内容保存。所以修改后的代码如下：

```
#!/bin/sh
exec scala -savecompiled "$0" "$@"
!#

println("Hello, iteblog!!")
args foreach println
```

运行这个脚本，我们可以发现第一次运行的速度还是很慢，但是当程序运行完，我们可以在同一目录下发现一个名为test.sh.jar的jar文件，这就是编译之后的class文件。第二次运行的时候，Scala会检查该脚本是否修改了，如果没有修改，会直接使用之前编译好的test.sh.jar文件，所以以后运行速度会很快：

```
[iteblog@www.iteblog.com iteblog]$ time sh test.sh
Hello, iteblog!!

real 0m0.298s
user 0m0.292s
sys 0m0.030s
```

我们来看看test.sh.jar里面的内容：

```
[iteblog@www.iteblog.com iteblog]$ jar -tvf test.sh.jar
 75 Wed Dec 16 19:17:32 CST 2015 META-INF/MANIFEST.MF
 832 Wed Dec 16 19:17:32 CST 2015 Main$$anon$1.class
 595 Wed Dec 16 19:17:32 CST 2015 Main.class
1097 Wed Dec 16 19:17:32 CST 2015 Main$$anon$1$$anonfun$1.class
 547 Wed Dec 16 19:17:32 CST 2015 Main$.class
```

我们可以看出这些就是编译后的class文件，我们可以通过scala命令去运行里面的类：

```
[iteblog@www.iteblog.com iteblog]$ scala -cp test.sh.jar Main
Hello, iteblog!!
```

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#)（）