

Python很慢？Python之父一句话亮了

程序员开发手册 昨天

Python的运行速度确实没有C或者Java快，但是有一些项目正在努力让Python变得更快。

Python代码简洁干净，但是大家都知道Python运行起来相对较慢 --- 在CPU密集型的任务上慢于C、Java和Javascript（但是大多数服务都不是CPU密集型的） --- 但是有些团队希望Python是尽善尽美的，所以他们准备从内而外地提升Python的性能。

如果你想让Python在特定硬件上运行得快一点，你至少有两个选择，每个选择都有一些弊病：

- a. 你可以创建一个Python运行时的替代品，但是最后你会发现你重写了一遍CPython。
- b. 你可以重写现存的部分代码来利用一些性能优化的特性，缺点是程序员需要做更多的工作。

下面是五种已有的方案，帮助你提高Python的性能。

01.PyPy

在CPython的替代品中，PyPy是最显眼的那一个（比如Quora就在生产环境中使用它）。它也最有机会成为默认解释器，它和现存Python代码高度兼容。



PyPy使用适时编译来加速Python，这项技术Google也在使用，Google在V8引擎中使用它加速Javascript。最近的版本PyPy2.5增加了一些提升性能的特性，其中有一项很受欢迎，它集成了Numpy，Numpy之前也一直被用来加速Python的运行。

使用Python3的代码需要对应地使用PyPy3。PyPy目前只支持到Python3.2.5，对Python3.3的支持正在进行中。

02.Pyston

Pyston，由Dropbox资助，使用LLVM编译器架构来加速Python，同样的它也使用了适时编译。相比于PyPy，Pyston还处于早期阶段，它只支持Python的部分特性。Pyston把工作分成两个部分，一部分是语言的核心特性，另一部分是把性能提升到可接受的程度。Pyston距离可以在生产环境使用还有一段距离



03.Nuitka

除了替换Python运行时，有些团队尝试将Python代码转换为能够在本地高效运行的其他语言的代码。其中著名的项目是Nuitka--把Python代码转换为C++代码--虽然运行时还是依赖Python运行时。这样限制了它的可移植性，不过性能提升是可观的。长期规划中，Nuitka还准备让C语言能够调用Nuitka编译的Python代码，这样性能提升将更加明显。



04.Cython

Cython（Python的C语言扩展）是Python的超集，它能把Python代码编译成C代码，并与C和C++进行交互。它可以作为Python项目的扩展使用（重新性能要求高的部分），或者单独使用，不涉及传统的Python代码。缺点是你写的不是Python，所以需要手动迁移，缺乏可移植性。



据说，Cython提供了一些特性来让代码更高效，比如变量类型化，这本质上是C要求的。一些科学计算的包，如scikit-learn依赖Cython的一些特性来保持操作简洁快速。

05.Numba

Numba结合了上面几个项目的想法。学习了Cython，Numba也采用了部分加速的策略，只加速CPU密集型的任务；同时它又学习了PyPy和Pyston，通过LLVM运行Python。你可以用一个装饰器指定你要用Numba编译的函数，Numba继承Numpy来加速函数的执行，Numba不做适时编译，它的代码是预先编译的。



Python之父说：大部分觉得Python慢的应用都是没有正确地使用Python。对于CPU密集型的任务有多种方法来提升性能--使用Numpy来做计算，调用外部C代码，以及尽量避免GIL锁。由于GIL锁目前还无法被替代，所以有很多项目开始尝试一些短期可行的替代方案，当然这些方案也可能转变为长期的可选项。

译者：诗书塞外

原文：<http://dwz.date/duAA>

[阅读原文](#)

喜欢此内容的人还喜欢

Python 同步与异步及阻塞等

init研究生三年



Python简单框架介绍

浅谈Python



python基础

素人三毛

