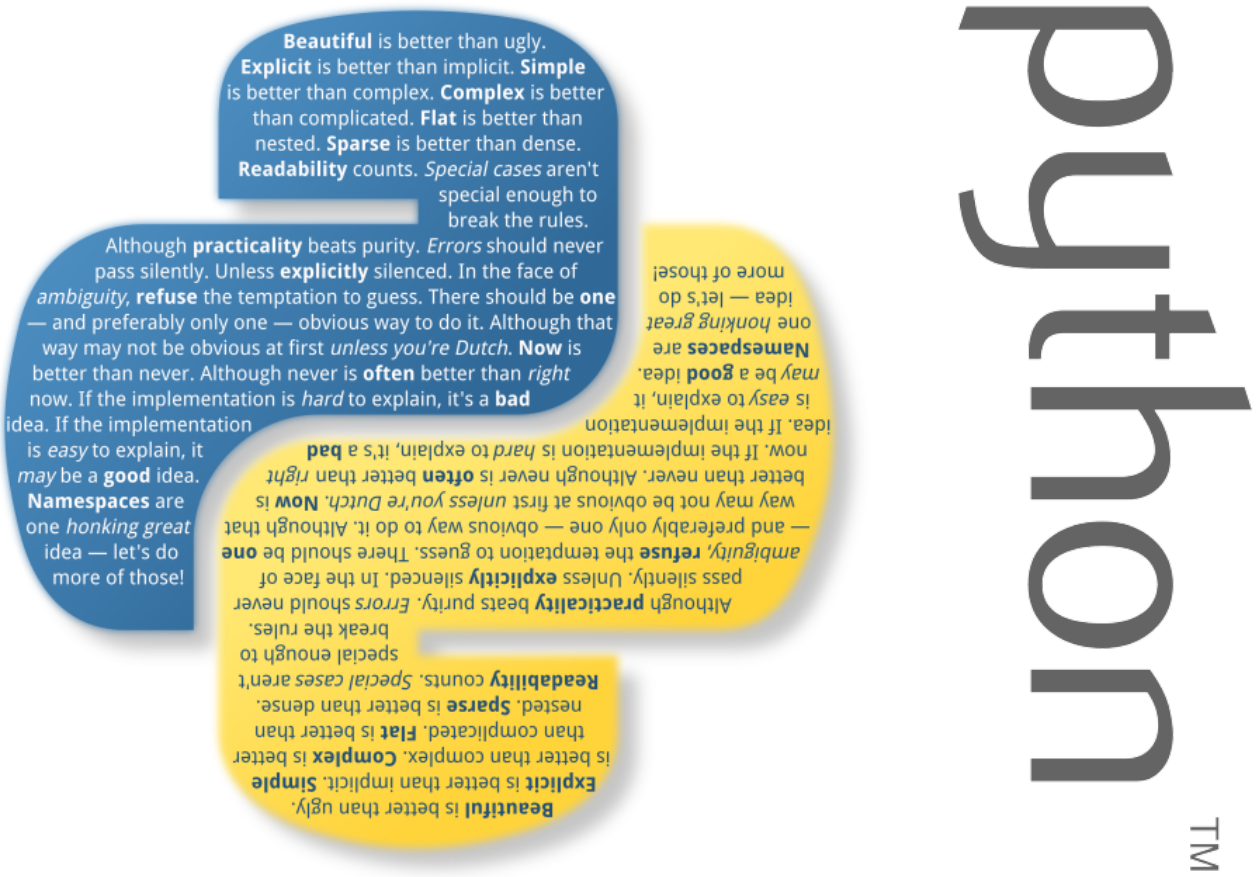
1004 words • 7~11 min read

大数据全栈式开发语言 – Python



前段时间，ThoughtWorks在深圳举办一次社区活动上，有一个演讲主题叫做“Fullstack JavaScript”，是关于用JavaScript进行前端、服务器端，甚至数据库（MongoDB）开发，一个Web应用开发人员，只需要学会一门语言，就可以实现整个应用。

受此启发，我发现Python可以称为大数据全栈式开发语言。因为Python在云基础设施，DevOps，大数据处理等领域都是炙手可热的语言。

领域	流行语言
云基础设施	**Python**, Java, Go

DevOps	**Python**, Shell, Ruby, Go
网络爬虫	**Python**, PHP, C++
数据处理	**Python**, R, Scala

就像只要会JavaScript就可以写出完整的Web应用，只要会Python，就可以实现一个完整的大数据处理平台。

云基础设施

这年头，不支持云平台，不支持海量数据，不支持动态伸缩，根本不敢说自己是做大数据的，顶多也就敢跟人说是做商业智能（BI）。

云平台分为私有云和公有云。私有云平台如日中天的OpenStack，就是Python写的。曾经的追赶者CloudStack，在刚推出时大肆强调自己是Java写的，比Python有优势。结果，搬石砸脚，2015年初，CloudStack的发起人Citrix宣布加入OpenStack基金会，CloudStack眼看着就要寿终正寝。

如果嫌麻烦不想自己搭建私有云，用公有云，不论是AWS，GCE，Azure，还是阿里云，青云，在都提供了Python SDK，其中GCE只提供Python和JavaScript的SDK，而青云只提供Python SDK。可见各家云平台对Python的重视。

提到基础设施搭建，不得不提Hadoop，在今天，Hadoop因为其MapReduce数据处理速度不够快，已经不再作为大数据处理的首选，但是HDFS和Yarn——Hadoop的两个组件——倒是越来越受欢迎。Hadoop的开发语言是Java，没有官方提供Python支持，不过有很多第三方库封装了Hadoop的API接口（pydoop，hadoopy等等）。

Hadoop MapReduce的替代者，是号称快上100倍的Spark，其开发语言是Scala，但是提供了Scala，Java，Python的开发接口，想要讨好那么多用Python开发的数据科学家，不支持Python，真是说不过去。HDFS的替代品，比如GlusterFS，Ceph等，都是直接提供Python支持。Yarn的替代者，Mesos是C++实现，除C++外，提供了Java和Python的支持包。

DevOps

DevOps有个中文名字，叫做开发自运维。互联网时代，只有能够快速试验新想法，并在第一时间，安全、可靠的交付业务价值，才能保持竞争力。DevOps推崇的自动化构建/测试/部署，以及系统度量等技术实践，是互联网时代必不可少的。

自动化构建是因应用而易的，如果是Python应用，因为有setuptools, pip, virtualenv, tox,

flake8等工具的存在，自动化构建非常简单。而且，因为几乎所有Linux系统都内置Python解释器，所以用Python做自动化，不需要系统预安装什么软件。

自动化测试方面，基于Python的Robot Framework企业级应用最喜欢的自动化测试框架，而且和语言无关。Cucumber也有很多支持者，Python对应的Lettuce可以做到完全一样的事情。Locust在自动化性能测试方面也开始受到越来越多的关注。

自动化配置管理工具，老牌的如Chef和Puppet，是Ruby开发，目前仍保持着强劲的势头。不过，新生代Ansible和SaltStack——均为Python开发——因为较前两者设计更为轻量化，受到越来越多开发这的欢迎，已经开始给前辈们制造了不少的压力。

在系统监控与度量方面，传统的Nagios逐渐没落，新贵如Sensu大受好评，云服务形式的New Relic已经成为创业公司的标配，这些都不是直接通过Python实现的，不过Python要接入这些工具，并不困难。

除了上述这些工具，基于Python，提供完整DevOps功能的PaaS平台，如Cloudify和Deis，虽未成气候，但已经得到大量关注。

网络爬虫

大数据的数据从哪里来？除了部分企业有能力自己产生大量的数据，大部分时候，是需要靠爬虫来抓取互联网数据来做分析。

网络爬虫是Python的传统强势领域，最流行的爬虫框架Scrapy，HTTP工具包urlib2，HTML解析工具beautifulsoup，XML解析器lxml，等等，都是能够独当一面的类库。

不过，网络爬虫并不仅仅是打开网页，解析HTML这么简单。高效的爬虫要能够支持大量灵活的并发操作，常常要能够同时几千甚至上万个网页同时抓取，传统的线程池方式资源浪费比较大，线程数上千之后系统资源基本上就全浪费在线程调度上了。Python由于能够很好的支持协程（Coroutine）操作，基于此发展起来很多并发库，如Gevent，Eventlet，还有Celery之类的分布式任务框架。被认为是比AMQP更高效的ZeroMQ也是最早就提供了Python版本。有了对高并发的支持，网络爬虫才真正可以达到大数据规模。

抓取下来的数据，需要做分词处理，Python在这方面也不逊色，著名的自然语言处理程序包NLTK，还有专门做中文分词的Jieba，都是做分词的利器。

数据处理

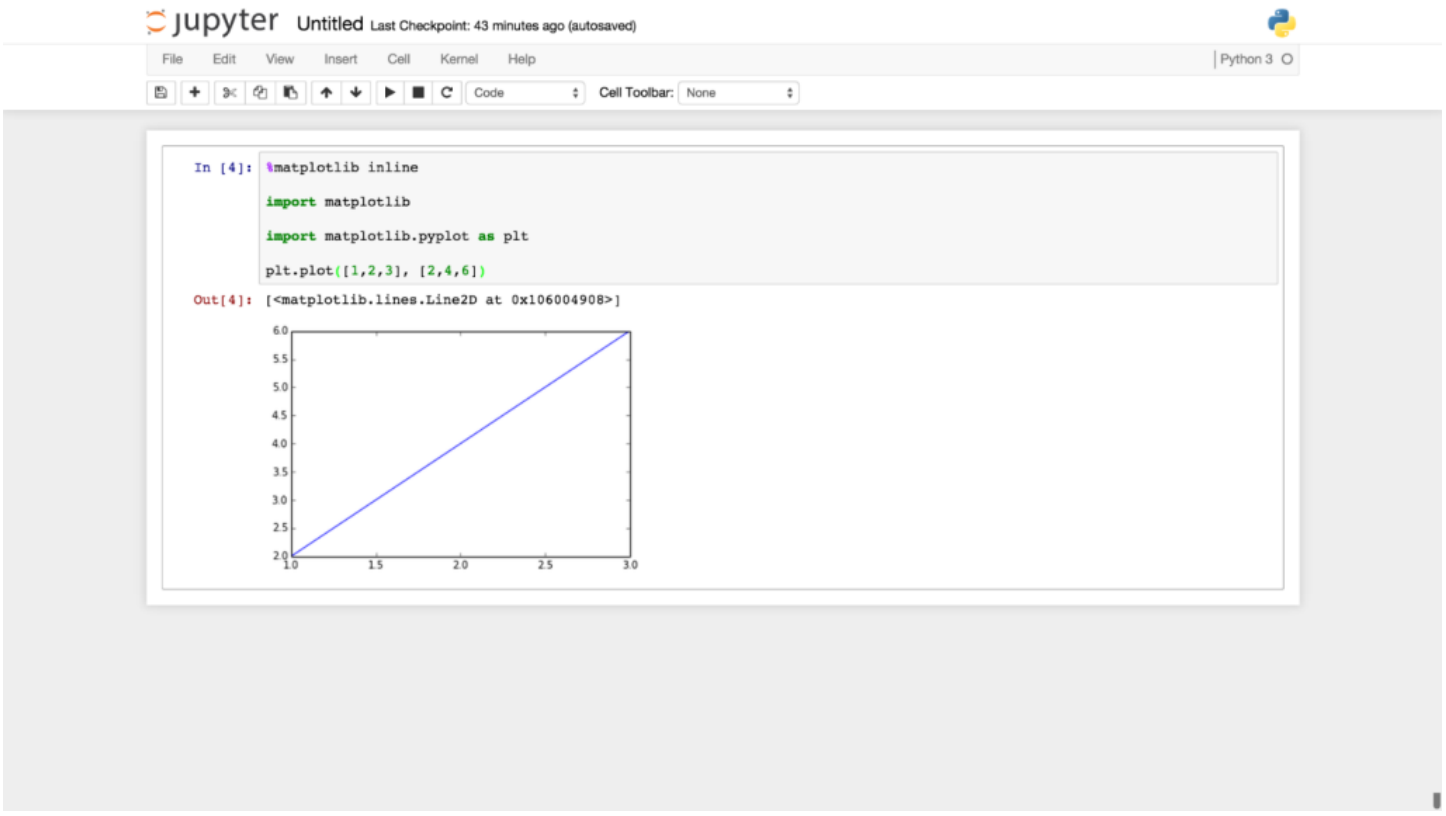
万事俱备，只欠东风。这东风，就是数据处理算法。从统计理论，到数据挖掘，机器学习，再到最近几年提出来的深度学习理论，数据科学正处于百花齐放的时代。数据科学家们都用什么编程？

如果是在理论研究领域，R语言也许是最受数据科学家欢迎的，但是R语言的问题也很明显，因为是统计学家们创建了R语言，所以其语法略显怪异。而且R语言要想实现大规模分布式系统，还需要很长一段时间的工程之路要走。所以很多公司使用R语言做原型试验，算法确定之后，再翻译成工程语言。

Python也是数据科学家最喜欢的语言之一。和R语言不同，Python本身就是一门工程性语言，数据科学家用Python实现的算法，可以直接用在产品中，这对于大数据初创公司节省成本是非常有帮助的。正式因为数据科学家对Python和R的热爱，Spark为了讨好数据科学家，对这两种语言提供了非常好的支持。

Python的数据处理相关类库非常多。高性能的科学计算类库NumPy和SciPy，给其他高级算法打了非常好的基础，matploglib让Python画图变得像Matlab一样简单。Scikit-learn和Milk实现了很多机器学习算法，基于这两个库实现的Pylearn2，是深度学习领域的重要成员。Theano利用GPU加速，实现了高性能数学符号计算和多维矩阵计算。当然，还有Pandas，一个在工程领域已经广泛使用的大数据处理类库，其DataFrame的设计借鉴自R语言，后来又启发了Spark项目实现了类似机制。

对了，还有iPython，这个工具如此有用，以至于我差点把他当成标准库而忘了介绍。iPython是一个交互式Python运行环境，能够实时看到每一段Python代码的结果。默认情况下，iPython运行在命令行，可以执行`ipython notebook`在网页中运行。用matplotlib绘制的图可以直接嵌入式的显示在iPython Notebook中。
iPython Notebook的笔记本文件可以共享给其他人，这样其他人就可以在自己的环境中重现你的工作成果；如果对方没有运行环境，还可以直接转换成HTML或者PDF。



为什么是Python

正是因为应用开发工程师、运维工程师、数据科学家都喜欢Python，才使得Python成为大数据系统的全栈式开发语言。

对于开发工程师而言，Python的优雅和简洁无疑是最大的吸引力，在Python交互式环境中，执行`import this`，读一读Python之禅，你就明白Python为什么如此吸引人。Python社区一直非常有活力，和NodeJS社区软件包爆炸式增长不同，Python的软件包增长速度一直比较稳定，同时软件包的质量也相对较高。有很多人诟病Python对于空格的要求过于苛刻，但正是因为这个要求，才使得Python在做大型项目时比其他语言有优势。OpenStack项目总共超过200万行代码，证明了这一点。

对于运维工程师而言，Python的最大优势在于，几乎所有Linux发行版都内置了Python解释器。Shell虽然功能强大，但毕竟语法不够优雅，写比较复杂的任务会很痛苦。用Python替代Shell，做一些复杂的任务，对运维人员来说，是一次解放。

对于数据科学家而言，Python简单又不失强大。和C/C++相比，不用做很多的底层工作，可以快速进行模型验证；和Java相比，Python语法简洁，表达能力强，同样的工作只需要1/3代码；和Matlab，Octave相比，Python的工程成熟度更高。不止一个编程大牛表达过，Python是最适合作为大学计算机科学编程课程使用的语言——MIT的计算机入门课程就是使用的Python——因为Python能够让人学到编程最重要的东西——如何解决问题。

顺便提一句，微软参加2015年PyCon，高调宣布提高Python在Windows上的编程体验，包括Visual Studio支持Python，优化Python的C扩展在Windows上的编译等等。脑补下未来Python作为Windows默认组件的场景。

新兴技术 | Python, 佟达 | 2015年7月16日 • pm9:01