

## Go语言，Docker和新技术

2017-10-26 陈皓



Go语言，Docker和新技术

朗读人：柴巍 12'00" | 5.50M

上个月，作为 Go 语言的三位创始人之一，Unix 老牌黑客罗勃·派克（Rob Pike）在新文章“Go: Ten years and climbing”中，回顾了一下 Go 语言的发展过程。其中提到，Go 语言这十年的迅猛发展大到连他们自己都没有想到，并且还成为了云计算领域中新一代的开发语言。还提到了，中国程序员对 Go 语言的热爱完全超出了他们的想象，甚至他们都不敢相信是真的。

这让我想起，我在 2015 年 5 月份拜访 Docker 公司在湾区的总部时，Docker 负责人也和我表达了相似的感叹：他们完全没有想到居然中国有那么多人喜欢 Docker，而且还有这么多人在为 Docker 做贡献，这让他们感到非常意外。此外，还跟我说，中国是除了美国本土之外的另外一个如此喜欢 Docker 技术的国家，在其它国家都没有看到。

的确如他们所说，Go 语言和 Docker 这两种技术已经成为新一代的云计算技术，而且可以看到其发展态势非常迅猛。而中国也成为了像美国一样在强力推动这两种技术的国家。这的确是一件让人感到非常高兴的事，因为中国在跟随时代潮流这件事上已经做得非常不错了。

然而，从 2014-2015 年我在阿里推动 Docker 和 Go 语言的痛苦和失败过程中，以及这许多年来，有很多很多人问我是否要学 Go 语言，是否要学 Docker，Go 和 Docker 是否能用在生产线上，这些问题看来，对于 Go 语言和 Docker 这两种技术，在国内的技术圈中有相当大的一部分人和群体还在执观望或是不信任的态度。

所以，我想写这篇文章，从两个方面来论述一下我的观点和看法。

- 一个方面，为什么 Go 语言和 Docker 会是新一代的云计算技术。
- 另一个方面，作为技术人员，我们如何识别什么样的新技术会是未来的趋势。

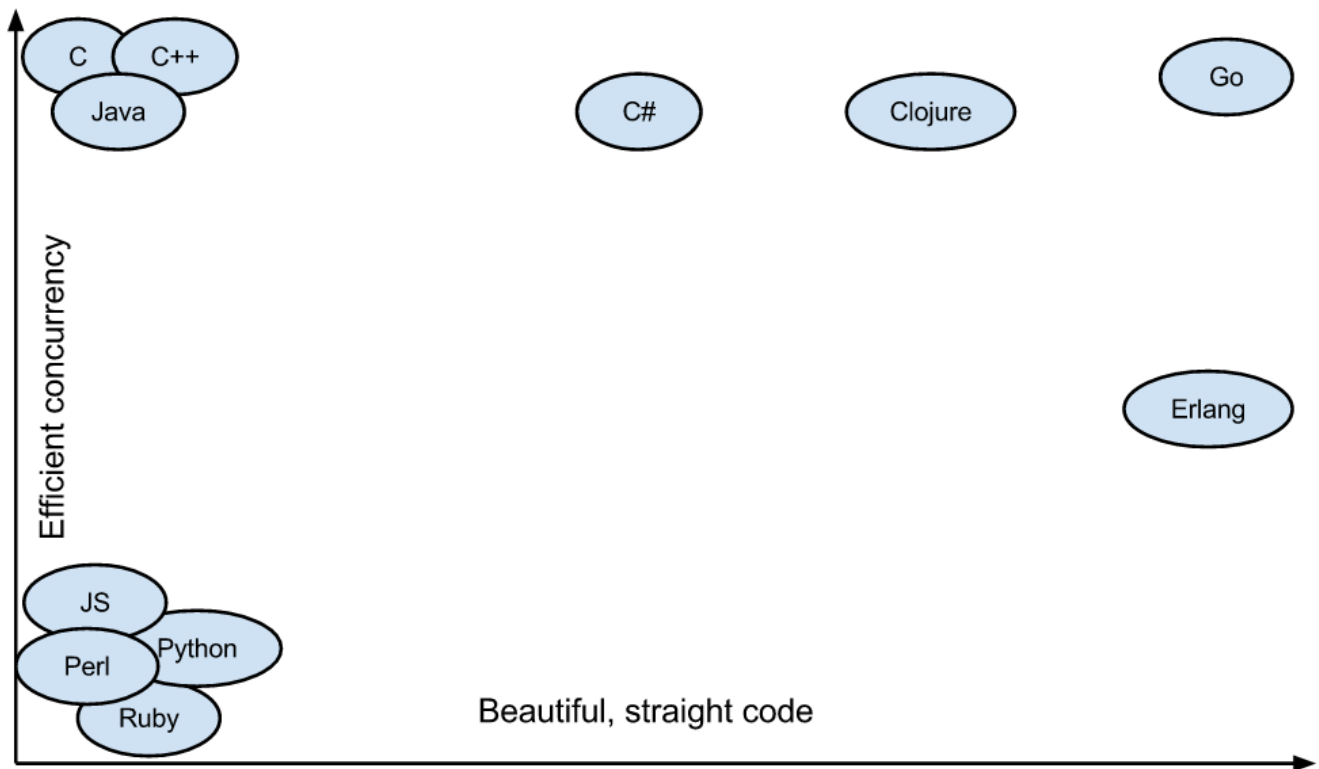
这两个问题是相辅相成的，所以我会把这两个问题揉在一起谈。

虽然 Go 语言是在 2009 年底开源的，但我是从 2012 年才开始接触和学习 Go 语言的。我只花了一个周末两天的时间就学完了，而且在这两天，我还很快地写出了一个能工作很好的网页爬虫程序，以及一个简单的高并发文件处理服务，用于提取前面抓取的网页的关键内容。这两个程序都很简单，总共才写了不到 500 行代码。

我当时对 Go 语言有几点体会。

第一，语言简单，上手快。Go 语言的语法特性简直是太简单了，简单到你几乎玩不出什么花招，直来直去的，学习曲线很低，上手非常快。

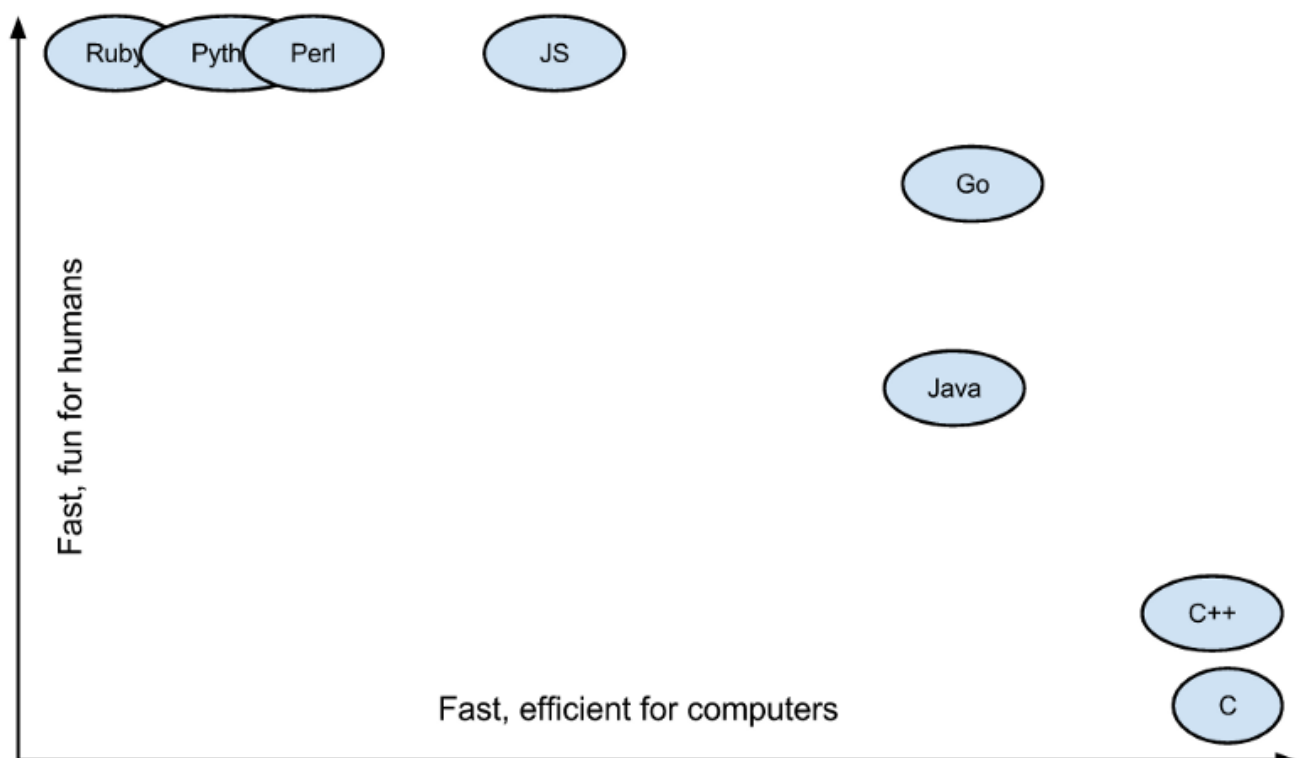
第二，并行和异步编程几乎无痛点。Go 语言的 Goroutine 和 Channel 这两个神器简直就是并发和异步编程的巨大福音。像 C、C++、Java、Python 和 JavaScript 这些语言的并发和异步方式太控制就比较复杂了，而且容易出错，而 Go 解决这个问题非常地优雅和流畅。这对于编程多年受尽并发和异步折磨的我来说，完全就是让我眼前一亮的感觉。



( 图片来自 Medium : Why should you learn Go? )

第三，Go 语言的 lib 库麻雀虽小五脏俱全。Go 语言的 lib 库中基本上有绝大多数常用的库，虽然有些库还不是很好，但我觉得不是问题，因为我相信在未来的发展中会把这些问题解决掉。

第四，C 语言的理念和 Python 的姿态。C 语言的理念是信任程序员，保持语言的小巧，不屏蔽底层且底层友好，关注语言的执行效率和性能。而 Python 的姿态是用尽量少的代码完成尽量多的事。于是我能够感觉到，Go 语言想要把 C 和 Python 统一起来，这是多棒的一件事啊。



( 图片来自 Medium : Why should you learn Go? )

所以，即便 Go 语言存在诸多的问题，比如垃圾回收、异常处理、泛型编程等，但相较于上面这几个优势，我认为这些问题都是些小问题。于是就毫不犹豫地入坑了。

当然，一个技术能不能发展起来，关键还要看三点。

- 有没有一个比较好的社区。像 C、C++、Java、Python 和 JavaScript 的生态圈都是非常丰富和火爆的。尤其是有很多商业机构参与的社区那就更为人气爆棚了，比如 Linux 的社区。
- 有没有一个工业化的标准。像 C、C++、Java 都是有标准化组织的。尤其是 Java，其在架构上还搞出了像 J2EE 这样的企业级标准。
- 有没有一个或多个杀手级应用。C、C++ 和 Java 的杀手级应用不用多说了，就算是对于 PHP 这样还不能算是一个好的编程语言来说，因为是 Linux 时代的第一个杀手级解决方案 LAMP 中的关键技术，所以，也发展起来了。

上述的这三点是非常关键的，新的技术只需要占到其中一到两点就已经很不错了，何况有的技术，比如 Java，是三点全占到了，所以，Java 的发展是如此好。当然，除了上面这三点重要的，还有一些其它的影响因素，比如：

- 学习曲线是否低，上手是否快。这点非常重要，C++ 在这点上越做越不好了。
- 有没有一个不错的提高开发效率的开发框架。如：Java 的 Spring 框架，C++ 的 STL 等。
- 是否有一个或多个巨型的技术公司作为后盾。如：Java 和 Linux 后面的 IBM、Sun.....
- 有没有解决软件开发中的痛点。如：Java 解决了 C 和 C++ 的内存管理问题。

用这些标尺来量一下 Go 语言，我们可以清楚地看到：

- Go 语言容易上手；
- Go 语言解决了并发编程和写底层应用开发效率的痛点；
- Go 语言有 Google 这个世界一流的技术公司在后面；
- Go 语言的杀手级应用是 Docker，而 Docker 的生态圈在这几年完全爆棚了。

所以，Go 语言的未来是不可限量的。当然，我个人觉得，Go 可能会吞食很多 C、C++、Java 的项目。不过，Go 语言所吞食主要的项目应该是中间层的项目，既不是非常底层也不会是业务层。

也就是说，Go 语言不会吞食底层到 C 和 C++ 那个级别的，也不会吞食到高层如 Java 业务层的项目。Go 语言能吞食的一定是 PaaS 上的项目，比如一些消息缓存中间件、服务发现、服务代理、控制系统、Agent、日志收集等等，没有复杂的业务场景，也到不了特别底层（如操作系统）的中间平台层的软件项目或工具。而 C 和 C++ 会被打到更底层，Java 会被打到更上层的业务层。这是我的一个判断。

好了，我们再用上面的标尺来量一下 Go 语言的杀手级应用 Docker，你会发现基本是一样的。

- Docker 上手很容易。
- Docker 解决了运维中的环境问题以及服务调度的痛点。
- Docker 的生态圈中有大公司在后面助力。比如 Google。
- Docker 产出了工业界标准 OCI。
- Docker 的社区和生态圈已经出现像 Java 和 Linux 那样的态势。
- .....

所以，早在 3、4 年前我就觉得 Docker 一定会是未来的技术。虽然当时的坑儿还很多，但是，相对于这些大的因素来说，那些小坑儿都不是问题。只是需要一些时间，这些小坑儿在未来 5-10 年就可以完全被填平了。

同样，我们可以看到 Kubernetes 作为服务和容器调度的关键技术一定会是最后的赢家。这点我在去年初就能够很明显地感觉到了。

关于 Docker 我还想多说几句，这是云计算中 PaaS 的关键技术，虽然，这世上在出现 Docker 之前，几乎所有的要玩公有 PaaS 的公司和产品都玩不起来，比如：Google 的 GAE，国内的各种 XAE，如淘宝的 TAE，新浪的 SAE 等。但我还是想说，PaaS 是一个被世界或是被产业界严重低估的平台。

PaaS 层是承上启下的关键技术，任何一个不重视 PaaS 的公司，其技术架构都不可能让这家公司成长为一个大型的公司。因为 PaaS 层的技术主要能解决下面这些问题。

- 软件生产线的问题。持续集成和持续发布，以及 DevOps 中的技术必需通过 PaaS。
- 分布式服务化的问题。分布式服务化的服务高可用、服务编排、服务调度、服务发现、服务路由，以及分布式服务化的支撑技术完全是 PaaS 的菜。
- 提高服务的可用性 SLA。提高服务可用性 SLA 所需要的分布式、高可用的技术架构和运维工具，也是 PaaS 层提供的。
- 软件能力的复用。软件工程中的核心就是软件能力的复用，这一点也完美地体现在 PaaS 平台的技术上。

老实说，这些问题的关键程度已经到了能判断一家依托技术的公司的研发能力是否靠谱的程度。没有这些技术，依托技术拓展业务的公司几乎不可能发展得规模很大。

在后面，我会另外写几篇文章详细地讲一下分布式服务化和 PaaS 平台的重要程度。

最后，我还要说一下，为什么要早一点地进入这些新技术，而不是等待这些技术成熟了后再进入。原因有这么几个。

- 技术的发展过程非常重要。我进入 Go 和 Docker 的技术不能算早，但也不算晚，从 2012 年学习 Go，到 2013 年学习 Docker 到今天，我清楚地看到了这两种技术的生态圈发展过程。让我收获最大的并不是这些技术本身，而是一个技术的变迁和行业的发展。

从中，我看到了非常具体的各种思潮和思路，这些东西比起 Go 和 Docker 来说更有价值。因为，这不但让我重新思考我已掌握的技术以及如何更好地解决已有的问题，而且还让我看到了未来。我不但有了技术优势，而且这些知识还让我的技术生涯多了很多的可能性。

- 这些关键新技术，可以让你拿到技术的先机。这些对一个需要技术领导力的个人或公司来说都是非常重要的。

一个公司或是个人能够占有技术先机，就会比其它公司或个人有更大的影响力。一旦未来行业需求引爆，那么这个公司或是个人的影响力就会形成一个比较大的护城河，并可以快速地产生产经济利益。

近期，在与中国移动、中国电信以及一些股份制银行进行交流的过程中，我已看到通讯行业、金融行业对于 PaaS 平台的理解已经超过了互联网公司，而我近 3 年来在这些技术上的研究让我也从中受益匪浅。

所以，Go 语言和 Docker 作为 PaaS 平台的关键技术前途是无限的，我很庆幸赶上了这个浪潮，也很庆幸在 3 年前我就看到了这个趋势，现在我也在用这些技术开发相关的技术产品，助力于为高速成长的公司提供这些关键技术。



左耳听风

洞悉技术的本质  
享受科技的乐趣

极客时间  
重拾极客精神·提升技术认知

陈皓

资深技术专家  
骨灰级程序员

扫码订阅

版权归极客邦科技所有，未经许可不得转载

精选留言





沉思猿

👍 26

正好最近两天在学习Go，陈老师的这篇文章来的非常及时！我的第一个Go程序是找到从维基百科A词条到B词条的路径，相当于网络爬虫。最大的感受就是像文章里提到的，用Go进行并发编程比Java轻松太多，thread、executor service、lock等都被抽象掉，仅仅通过goroutine和channel就可以完成很多工作。

期待关于微服务和PaaS的深度文章！

2017-10-26



匹诺曹

👍 15

php是世界上最好的编程语言😊

2017-10-26



李志博

👍 9

这个专栏真不错，极客时间就相当于程序员版的得到，这才是我们程序员需要的

2017-10-26



菡萏如佳人

👍 7

GO语言，Let's go!

2017-10-26



匹诺曹

👍 6

谢谢陈皓老师，我一定会好好学习，每天精进，我会在未来的留言里记录我的学习心得。

2017-10-26



neohope

👍 3

陈老师，您好！我想请教一个问题，按我理解docker容器和宿主机是共享内核的，容器之间进行了资源隔离，我自己也试过从空白建立一个镜像。但我对linux没有很深的了解，有一个问题一直没想明白，docker是如何使用同一个宿主的内核，兼容各linux发行版的呢？按我理解，内核和各容器中gcc库版本都不一致，应该很难保证兼容性的。否则我在一个发行版下编译的c程序，到另一个发行版下，就不需要重新编译了，但现实却并非如此。如果您比较忙，能否告诉我一下关键字，我自己去查一下，感谢！

2018-01-07

作者回复

可以参看我coolshell中docker的相关文章

2018-01-07



killua

👍 3

听到生态圈，社区这些名词总觉得是高级码农的上层社交圈。是说我的级别还太low么

2017-10-26



Lincoln

👍 3

前几天在mac上搭了 go 的开发环境，写下了 go 的第一行 hello world 😊

2017-10-26



红猪

👍 2

从区块链入坑go和docker，刚开始是抗拒的，本来概念就难理解，还整一堆新技术作为学习障碍，现在因为项目需要逼着学了，很想知道皓哥是怎么在短时间快速get到精华的，对于新手程序员，感觉哪儿都是重点

2018-02-23



天真有邪

👍 1

请问下，java4年多了现在是学习c，c++还是go

2018-03-10



blackpiglet

👍 1

陈老师能不能讲一下如何进入到开源项目中的经验，现在我正在做ceph相关的事情，能想到的方法就是订阅邮件，看issue tracker，看会议记录等。有没有其他更好的方法呢？

2018-01-11



总指挥

👍 1

老师，可以推荐下go的入门书籍吗？

2018-01-07

| 作者回复

网上有很多教程，自行Google吧

2018-01-09



古岩

👍 1

请问大神，对区块链技术的发展和未来，是怎么看的呢

2017-12-30



、、cryAllen

👍 1

赞，准备入坑中。

2017-11-04



噜噜猪

👍 1

耗子兄，有个问题想咨询，我现在在创业公司做开发，独立做一整套系统，系统多方面涉及数学知识，导致需要给市场人员讲解原理，可是我觉着最近脾气越来越差了，因为讲第一遍时候，会和和气气，但是他们听不懂，然后讲第二遍他们还听不懂，第三遍我就会很恼火，实在不知道该如何调节，希望能解答

2017-10-29



max

👍 1

更新速度有点慢哎，不过瘾

2017-10-27



幻影霸者

👍 0

表示go写着好难受，于是弃坑了



2018-05-21



一只傻白兔

👍 0

看完文章，我有个问题。Python将何去何从？

2018-05-02



晓聪

👍 0

现在许多区块链项目都采用go语言，为了研究以太坊看来要入手go了。

2018-04-29



雄哼哼

👍 0

go语言主要用于中间件，目前我了解到有部分公司开始使用go语言开发后台业务逻辑，go语言在业务开发这块的前途有吗？

2018-04-11

作者回复

微服务方面可以

2018-04-12