Rust今天4岁啦, 为什么越来越多的知名项目用Rust来开发？ - 云+社区 - 腾讯云

https://cloud.tencent.com/developer/article/1428784

区块链大本营 Sat Dec, 19 06:07

作者 | Mike Tang

责编 | Aholiab

出品 | 区块链大本营（blockchain_camp）

4年前的今天（2015年5月15日），Rust编程语言核心团队正式宣布发布Rust 1.0版本。

4年来，它优雅的解决高并发和高安全性系统问题的能力，受到了越来越多开发者的喜爱。**并且连续4年，在Stack Overflow开发者「最受喜爱编程语言」评选中获得第一名**。

近来，越来越多的著名区块链项目已经选择使用Rust作为其开发语言，包括：Parity、Polkadot、Substrate、Grin、Ethereum经典、Holochain、Cardano-Rust、Exonum、Lighthouse、Nimiq、Nervos、Conflux-Rust、Codechain、Witnet等众多知名度较高的项目。

本文试图探讨这一种趋势出现背后的原因。并且系统的讲解Rust语言对于区块链开发的影响和其优势。

Rust：一个安全和并发的软件系统

Rust 是一门系统级编程语言，被设计为保证内存和线程安全，防止段错误产生。

作为系统级编程语言，它的基本理念是“零开销抽象”。**理论上来说，它的速度与 C / C++ 同级**。Rust 可以被归为通用的、多范式、编译型的编程语言，类似 C/C++。

与这两门编程语言不同的是，**Rust 是线程安全的！它的目标是，创建一个安全和并发的软件系统**。Rust强调安全性、并发和内存控制。**尽管 Rust 借用了 C/C++ 的语法，却杜绝了空指针和悬挂指针**，而这二者是 C/C++ 中系统崩溃、内存泄露和不安全代码的根源。

虽然 Rust 是一门系统级编程语言，但并不意味着它只能写底层程序（操作系统、驱动、工具、数据库、搜索引擎等），**它的抽象层次之高完全给人惊艳的感觉**，实践证明它对问题建模的能力和方便性不比 C++/Java/Python/Ruby 差。

但 Haskell 这类超高抽象语言，也不是 Rust 的发展方向。Rust 力求在抽象与现实世界中找到一个平衡。

目前，Rust 已经在 StackOverflow 的年度语言评选中，连续 4 年荣获“程序员最喜爱语言”第一名。

Rust语言的历史

Rust 最早是 Mozilla 雇员 Graydon Hoare 的一个个人项目，从 2009 年开始，得到了 Mozilla 研究院的支助，2010 年项目对外公布。

2010 ～2011 年间实现的自举。从此以后，Rust 经历了巨大的设计变化和反复（历程极其艰辛），终于在 2015 年 5 月 15日发布了 1.0 版。

在这个研发过程中，Rust 建立了一个强大活跃的社区，形成了一整套完善稳定的项目贡献机制（这是真正的可怕之处）。Rust 现在由 Rust 项目开发者社区维护。

下图描述了Rust的基因图谱：

Rust仍是小众语言

Rust从1.0算起，到目前（2019年5月）为止，已有长足的发展。但整体而言，比起，Python、C、C++ 等，Rust 的使用并不算特别广泛（毕竟太年轻）。

调查显示，Rust 使用率不高的很大一部分原因是选择 Rust 的公司占少数。

不过，Rust 在工作中的占比却越来越高。过去一年间，Rust 在商业应用上有着令人惊讶的成长。包括一些人们耳熟能详的公司：

- **亚马逊**：用 Rust 构建工具；
- **Facebook**：源代码控制工具；
- **Google**：作为 Fuchsia 项目的一部分；
- **微软**：在新的 Azure 物联网框架中部分使用 Rust；
- **Twitter**：用 Rust 作为构建团队支持的一部分。

国内的百度、阿里，也已经在内部小范围的使用上了Rust。相信，随着更多公司对 Rust 的选择，Rust 的成长速度将更加迅猛。

Gavin Wood把Rust带进区块链

第一个将Rust应用于著名区块链项目的应该是大名鼎鼎的Gavin Wood。Gavin Wood 博士在 2015 年下半年离开了以太坊社区，创立了 Parity Technologies，推出以太坊（Ethereum）客户端 Parity，成功占领以太坊社区的半壁江山。**而这个Parity客户端就是使用Rust写的**。

实际在 Parity 出现之前，MaidSafe项目就已经出现了。**MaidSafe用 Rust 语言尝试了很多东西**。Rust语言本身强调的安全性以及MaidSafe这些前沿项目，可能给了 Gavin 充分的理由选择 Rust 作为 parity 开发语言。

Rust本身的语言特性（安全、高性能、并发编程）与区块链的特性（分布式、加密、安全敏感）天生有相当大的重合性。

不仅如此，Rust作为一个编程语言显得不同，不是因为它的语法多么漂亮（实际有些人甚至觉得有点丑）或者社区多么受人欢迎，而是因为当用它写代码时获得的那种信心。

看起来Rust会影响你写代码的的效率和表达力，但令人相当惊奇的是，结论完全相反：写一个有效率的、符合习惯的Rust程序比写一个有潜在危险的程序容易得多。

下面是Linux内核在2018年一月到四月期间发现的bug：

而对于Rust而言，上图右侧占比 51% 的部分，从语言层面就可以避免。也就是说，对Rust来说，根本不存在上图右边这些问题。

目前有两种观点：“一个人只需要知道如何写C”和“只需要把最底层的东西留给专业人士”，这两种观点都是不够的。Linux内核是由强中强的程序员写的，可能是工业界最前面的5%的程序员，但是，仍然，年复一年地搞出 CVE 来。

你有可能会觉得几百万行代码中出现50个bug不算什么。但在一些关键领域，出bug就意味着系统性风险（如，心脏起博器中的bug会导致生命危险）。而且，这50个bug是我们已经找到的，谁知道我们还有多少个bug还没找到？而用了Rust，我们可以事先知道答案。

厉害的是，Rust在实现内存安全和并发安全的同时，并没有以损失性能为代价。更牛逼的是，它甚至是用同一套抽象解决了内存安全和数据竞争这两个不同领域的问题。

Rust的零开销抽象让你在享受安全性的同时，又不损失性能。这正是传统的程序员梦寐以求的。

以Parity为例，Parity使用Rust，正是因为用Rust写复杂和高性能的代码时，不用担惊受怕。用Rust写程序，远离未定义行为、数据竞争和内存安全问题。更别说，Rust运行速度快，写起来有趣，易读，还几乎没有运行时。

内存安全问题如此困难，因为你无法容易地写出测试来捕获它们。如果你在beta阶段之前没有找出bug，那么这个bug就可能会在代码中呆几年，就像一个读秒的定时炸弹。当然，也有Valgrind这类的工具，来辅助你捕获这些bug。但是如果在执行时，没有触发内存问题，或这类工具生成的代码在运行时，没有执行，那么它们也捕获不到。

所以，通过使用Rust，我们消除了一大类最复杂和最不可预测的错误。

Rust中内嵌了形式化证明理论，不过仅限于对内存安全和并发。Rust在内部用逻辑证明了你的程序是正确的。这也就是你为什么写Rust代码会写得如此有信心的原因。你的每一次编译，都有一组数学理论在为你提供证明服务。

在很长一段时间内，像Haskell这种函数式编程语言的一大杀手级特性就是可以比较容易地进行形式化证明，而这对于传统的命令式编程语言是不可能的（因为有共享可变性，不安全的指针运算，和不可控的副作用）。

但Rust的出现，改变了这种情况，作为一个命令式语言，它却走在被证明的路上。到目前为止，标准库的一部分已经被证明是正确的。

那些用Rust开发的知名区块链项目

parity

说起知名的用Rust写的区块链项目，Parity首当其冲。Parity是一个以太坊节点客户端parity-ethereum。

Parity Technologies 是他们公司的名称。现在 Parity 旗下已经包含了一套客户端和各种库，包括：

- **parity-ethereum；**
- **parity-zcash；**
- **parity-bitcoin；**
- **shasper。**

也就是说，Parity Technologies 给使用Rust开发区块链提供了一套「全家桶」。

polkadot/substrate

Polkadot 是由Web3基金会发起的一项计划，由 Parity Technologies 负责开发，旨在却不限于使区块链互联。

Polkadot 使开发者和企业能够利用其协议建立区块链，即平行链（parachain）。只要这些平行链建立在Polkadot的基础之上，它们将共享同样的权威证明（PoA）共识。

由于该类型共识嵌于 Polkadot 中，平行链开发者可以专注于各自区块链的特异性。所有平行链都和一种被称为中继链（relay chain）的通用区块链无缝连接，后者扮演连接所有平行链的角色。

Polkadot 项目由Gavin Wood主导推进，其实跟上面的parity全家桶属于同一家公司，但属于不同的团队在做。

如果你实时关注了 Polkadot 的进展状况，你可能会经常看到“Substrate”这个词。它是 Polkadot 项目的重要组成部分。Parity Substrate 是独立于 Polkadot 的项目。Polkadot 基于 Substrate ，其它基于 Substrate 的项目也能在 Polkadot 网络上运行（形成一个生态）。

那么什么是 Substrate 呢？你可以将其看作类似于 Express 或其他 Web 应用程序的框架，但它是用于构建分布式或去中心化的系统的框架，可以构建例如加密货币项目，或消息总线系统。

正如大多数 Web 应用程序不需要重新实现自己的 HTTP 协议一样，对于每一个团队创建新链时，也不需要从头实现网络和共识的代码，这浪费精力。

更不用提为了实现业务逻辑，必须雇用密码学家、安全研究员、网络工程师、开发人员（以协调更新）等等了。如果你使用 Substrate 来构建一个新项目，只需要在代码中实现少量的函数回调，就可以轻松快速创建一条链。

参考链接：

https://github.com/paritytech/polkadot

https://github.com/paritytech/substrate

grin

grin是当今区块链界小网红。少数旨在成为真正货币的项目之一。

Grin 致力于提供隐私、可扩展性的加密货币，计划在 2019 年初发布。其几个独特的属性：使用名为 MimbleWimble 隐私保护区块链格式；只在区块链中存储少量数据，运行完整节点既便宜又容易。

Mimblewimble 由匿名人士开发，团队分布在世界各地；没有任何预挖的 PoW 机制，更平等和去中心化。

Grin 核心开发者、Grin Council 成员 D

grin的出现，预示着回归比特币的一股潮流的出现，也许，这才是这一类项目正确的方向。

参与gin的开发交流，相当令人愉悦。

holochain

Holochain致力于解决Dapp落地的事情，也就是说，它在构建一个云托管网络。

当人们都在研究如何把区块链性能提高的时候，Holochain却在做造汽车的事。有时候思维固化并非好事，换一种思维方式或许就可以找到解决问题的新路径。正如Holochain官网上的文字一样：**Think outside the blocks**。

Holochain这个项目很有意思，它不是狭义上的区块链。人们研究区块链本质上是在研究通过解决信任问题完成价值传输，从而完成生产关系改变。

为了达到最终目的，我们不一定要把眼光仅限于某种特定技术上。无论是区块链、DAG、Holochain，只要能达到我们最终目的，那么都可以认为是好的技术。

区块链中为了保证信息一致性，系统中每个节点都是一个单独的账本，这些账本记录同样的信息。虽然，区块链这种分布式账本技术满足了一致性，但却造成了巨大的信息冗余。

Holochain认为共识是可以分化的，没必要所有的事件都去共识。因此它创新的从另一个角度去解决共识问题。Holochain将一个账本分割成若干份，交给每一个参与节点，并通过DHT技术和密码学技术保证账本之间的一致性。

简单来说，我们可以把DHT的作用理解为拼图上的图画，我们可以通过图画来判断这个拼图是否完整。

节点不再需要多账本同步记录，只需要记录与自身相关的内容就可以达成全网共识，这就是Holochain的革新之处。

nervos

首先，Nervos 要做什么？官网上说，Nervos 是要：**为未来的加密经济构建分层的基础设施**。分层的方式也是现在区块链主流体系设计，Layer 1 发行原生资产，支持高安全性高去中心化，牺牲效率。

而 Layer 2 去做扩容的事情。那么 Nervos CKB 作为体系的Layer 1，主要有以下两个目的：资产的储存和协议的增强（安全性和去中心化）。

注意，这也是 Nervos CKB 经济体系设计的原则，即资产储存大于价值交换。

Nervos有几个有趣的设计亮点：

- 解决通缩与通胀的问题；
- 解决手续费的问题Nervos CKB 的一个理念是，智能合约平台的价值除了原生代币外，还应包含平台上发行的其他加密资产的价值；
- 解决资源的价格问题；
- 解决利率的发现与调整问题。

exonum

由 Bitbury 出品。是一套使用Rust开发的许可链开发框架。使用exonum，可以方便的搭建联盟链。

结束语

作为一个刚刚4岁的语言（今天2019年5月15日，是Rust语言正式发布4周年生日），Rust语言正在IT工业各个领域快速发展，而由于区块链本身的特质，区块链领域是较早接纳Rust的领域之一。

在区块链领域，Rust正以势如破竹之势占领区块链新兴项目市场，很多著名的老项目也在考虑转向使用Rust重写。

与此同时，WebAssembly技术的飞速发展带来的成果也及时地应用到了区块链中。Rust对WebAssembly提供了一等支持。可能是所有语言中目前为止最好的。具体请参考：

https://rustforce.net/article?id=6807cc27-0e6c-4bc9-baa3-8b0c68ae9529

相信，随着Rust语言的 async/await 特性在今年的稳定，使用Rust进行区块链开发会变成更加轻松方便。选择使用Rust作为第一开发语言的区块链项目也会越来越多，我们会迎来一波的Rust语言学习高潮。

谨以本文作为Rust 4周年的生日礼物，祝Rust生日快乐！

**关于作者：*

Mike Tang，资深程序员，Rust语言爱好者，从14年下半年接触Rust，Sapper Web框架主要作者。从 18 年开始聚焦于区块链领域的学习研究，参与贡献 cita，rust-libp2p，grin等项目。

参考资料：

https://rustforce.net.cn/section?id=17dd744e-429a-42b1-8d30-01d31d556ab0

https://www.jianshu.com/p/cca7e4f46b86

https://www.parity.io/why-rust/