

关于 **Scala** 学习和就业中若干问题的讨论

凤凰木 <weiwen@tubi.tv>

June 26, 2021

团队的需求

- 能用 Scala 解决问题
- 能在已有的 Scala 项目上工作
- 基本的算法，数据库，各级架构设计能力

开发者的需求

- 高大上的技术团队
- 天天写 Scala, 天天函数式, 与 ** 的 J***,G* 说拜拜
- 函数要写最纯的 (存疑)
- 类型系统要用最高级的 (存疑)
- 什么 Akka, Actor, Cats, ZIO, Monad, Applicative, Functor, Category Theory, 能上的都给它上去
- ...

国内 Scala 社区的问题

- 规模过小
- 整体技术能力还稍欠
- 有青黄不接的潜在问题
- 大部分号称写过 Scala 的人, 可能会被认为不会写 Scala, 尤其是从 Spark 转过来的开发者

开发者面临的问题

- Scala 技术栈的团队极少
- Scala 入门门槛和需要的投入，要么被高估，要么被低估
 - 高估，疯狂学习理论知识，一写代码就扑街
 - 低估，一写代码直接扑街
- 理论和实践的差距过大，缺乏行之有效的弥合路径
- Scala 岗 == 数据搬运岗，薪资低
- 群里有些老司机骂人特别凶，有问题也不太敢问

冯唐《不二》选段

这样苦口婆心都劝不走的，弘忍就焚香，在新来的脑袋上点十二个点，烧戒疤，让他们成为真正的和尚。

“若烧身、烧臂、烧指。若不烧身、臂、指供养诸佛，非出家菩萨。”本来，可以不烧十二个，可以只烧九个。弘忍偏要烧十二个，而且弘忍的香特别粗，十二个戒疤就是一脑袋的戒疤。烧一个新来的，整个东山寺人皮烧焦的味道，盘旋一整天。咬定受戒的人，烧小一半儿，烧到前六个戒疤，闻到自己皮肉的味道在周身盘旋，基本就一声惨叫，跑出山门。遇上晕倒的，弘忍从来不趁着他们没有知觉，将剩下的戒疤烧完。弘忍从来都是停下来，喝喝茶，枯坐一阵，等他们醒来。他们醒来，通常的第一个问题是，还要烧几个？

弘忍从来不说，没几个要烧了，就快烧完了。弘忍从来都说，还有好些个要烧呢。这些醒来的人，往往一声惨叫，也跑出山门。日子久了，东山寺附近聚居了不少人，头皮上有一到十一戒疤不等，人数比东山寺的僧人多好几倍。这些人中间，两点的看不起一点的，三点的看不起两点的，四点的看不起三点的，依此类推。但是有两个例外。一个例外是从一点到十点的，都看不起十一点的。一点到十点的，看到十一点的，先是强忍，沉默一小会儿，然后是忍不住地笑，“傻屁啊，哈，傻屁啊，哈，真是大傻屁啊，哈哈。”

一些积极的信号

- Scala 生态仍在进化，而且速度不慢
 - Scala 3 开始登上舞台
 - ZIO, cats 迅速发展
 - akka, alpakka 活跃演化中
 - sbt, mill 活跃演化中
- JVM 的发展，逐渐变得对函数式编程有利，比如有 ZGC 这种技术的出现
- Scala 仍是当前数据系统中难以或缺的角色
- Scala 社区低调而理性，藏龙卧虎
- 仍有一些像 Tubi, 水滴这样坚持 Scala 技术栈的团队和公司

我们热爱 Scala

关于门槛,

- 天赋不是必须的, 起点也不如预想的重要
- 实践优先于理论, 以产出为导向
- 范畴论, 函数式编程理论, 一开始都不是必需的
- 大多数不是 Scala 特有, 而是程序员成长的普遍问题

成为合格的工程师，而不是会写 Scala 的程序员

- 打好 Linux 基础
- 由简入繁，逐步学会设计
 - 函数的设计
 - Service API(高内聚，低耦合)
 - 数据存储结构的设计 (表，各种 NewSQL DB 的存储细节)
 - 由功能模块到子系统的架构设计
 - 始终保持清醒，学会用设计的思维去处理要实现的业务逻辑
- 学习数据中间件 (RMDBS, Kafka, Redis, Aerospike, etc.)
- 学习 JVM 基础知识
- 学习分布式系统理论

Scala 程序员应知必会

- Scala 基本语法
- Scala 标准库，主要是集合库和并发相关 (Future)
- 基本的函数式编程技能，掌握基本类型系统
- 能写出正确的并发和异步代码 (Scala Future)
- ScalaTest 和 Spec2 等测试库的用法
- 常见的生存相关类库
 - DB 类, slick, quill, doobie
 - HTTP 相关, play, akka-http, http4s
 - JSON 类, circe, json4s
 - Protobuf 类, scalapb, akka-grpc
 - 配置类, typesafe config, pureconfig
- akka (actor, stream, cluster, alpakka)
- 一点点的 cats
- 以上基本上涵盖了 Tobi 的 Scala 技术栈
- 相信也是大多数公司的招聘要求

一些建议

- 原则上禁用 `var`, `for`, `while`
- 禁止写出存在阻塞的代码
- 不要滥用 `actor`
- 不要滥用 `cluster`
- 用好 `Scala` 标准库（这是个宝藏）
- 尽量让代码函数式化，尽量满足引用透明的要求 (referential transparent)
- 合理抽象，但抽象的层次不宜过高，避免过度工程化
- 如有可能，多请人 `review` 自己写的代码，不要怕被喷

一些建议

- 理论结合实践，以代码产出为导向
- 不宜过度追求纯粹的函数式编程和复杂的高阶类型系统
- 算法练习（仅对语法和标准库的掌握有帮助）
 - S-99: Ninety-Nine Scala Problems
 - Leetcode 刷题
 - Euler Project (偏数论)
- 如果是初学，可以尝试写一些练手性的项目来提升代码技能
 - 写一个简单的个人博客系统
 - 实现一个分布式的缓存或者计数系统，比如用于流量控制（可以基于 akka）
 - 写一个简单的即时聊天系统
 - 写一个简单的交易所撮合引擎

谢谢大家！