

1. 引言

开源是一个比较时髦且有情怀的词汇，不少公司也把开源当做 KPI 或者是技术宣传的手段。没有开源就没有 Unix/Linux、Android、Git、MySQL、PHP、Go...现在几乎所有的软件或多或少直接和间接都依赖开源项目。没有人蠢到什么问题都要自己从头解决一遍。抵触开源就等于：不懂得依赖、不懂得分享也就是不懂得现代人的合作方式。正如侯捷所言：源码面前，了无秘密。但是这个世界又总是充满矛盾的，最顶级的项目有着技术壁垒，是一个公司赖以生存下去的保命石，故像Amazon Auraro的，Google自己的分布式文件系统，Spanner系统等又都是闭源的。开源和闭源看似是一种矛盾，却都是一个公司乃至一个国家战略上的博弈罢了。而开源的浪潮无疑是当下驱动软件发展的最强引擎。

2. 对开源的一些理解

2.1 什么是开源

简而言之，开源就是拥有源代码版权的人，允许其他人在一定许可证所述范围内，访问源代码，并用于一些自己的目的。最基本的要求就是其他人可以访问源代码，另外获取代码后能做什么，就需要一个专门的许可证来规范（可以是自己写的，也可以用一个人写好的）。里面一般会规定诸如对修改代码、新增代码、后续工作是否需要开源以及专利相关的事项。举一个最简单的例子，我们写一个 main.c 里面有一行 `printf "Hello World!"`，再和某个许可证文件一起扔到 GitHub 上，我们就有一个满足最低要求的开源项目了。

2.2 为什么要开源

很多人觉得代码是一个软件公司最宝贵的资产，把这些最宝贵的资产让别人免费获取，对你们有什么好处？如果对手拿走了你们的代码，另起炉灶和你们竞争怎么办？或者是用户直接获取源代码，用于自己的环境中，那你们如何收钱呢？对一个技术型公司来说，最宝贵的资产其实是人，对一个开源项目来说，最核心的资产是一个活跃的开源社区以及他人对这个项目的认可。我从下面几个方面谈一谈我对开源的一些理解。

- **Branding 很明显。**开源是一种非常好的 PR、Branding 的手段，大多数大公司做开源也是这个目的，可以以一种成本几乎为零的方式宣传企业名，树立技术型企业形象。
- **人才获取。**人才招聘最大的难处就是如何鉴别这个人的能力。如果这个人已经给你的项目贡献了一些代码，并且代码质量比较高、贡献过程中和你的沟通很顺畅，那么一方面说明这个人软硬实例都不错，另一方面说明这个人对你做的事情很有兴趣。
- **社区贡献。**无数的开源社区贡献者贡献自己的力量，共同维持这样一个互助互利的社区，支撑社会技术进步。我们也从开源社区中获得了许多支持。对基础软件的意义。
- **基础软件。**对于一个数据库这样的基础软件，最重要的就是正确性、稳定性和性能。只有当你的产品有足够多的人试用，甚至用于生产环境，才可能有足够多的问题反馈以及产品建议。

3. 如何参与开源项目

1. **试用**：最简单的参与方式是试用开源项目，这也是开源最大的一个好处，所有人都可以随时试用，相当于有很多人帮助项目作者做测试。
2. **报 Issue**：试用过程中大家可能会遇到各种问题，特别是文档中没有提及的问题，反馈问题的最佳方式是在 Github 上新建 Issue，这样所有的人都可以看到，而且通过 Issue 来反馈我们也会更重视一些，有人会定期扫一遍未处理的 Issue。在 Issue 中尽可能详细的描述清楚遇到的问题，以及一个可操作的复现步骤，包括所用 Binary 的版本、部署方式、客户端以及服务的日志、操作系统的日志（如 dmesg 的输出）。如果不能复现，也尽可能详细地提供 Log。这些对开发人员追踪 Bug 会非常有用。
3. **提出建议**：如果对项目有什么建议，也可以通过新建 Issue 来反馈，我们一般会给出是否会支持，如果要支持的话，大概会在什么时候支持。
4. **提 PR**：当使用过程中遇到问题或者需要新的 Feature，而觉得自己有能力 Fix 或者是当前官方还没有精力 Fix 时，可以尝试自己修改代码，解决问题。

4. 课程收获

这是第一次上也是第一次听说有专门针对**开源**的课程。课上老师的分享很多，也给自己带来了很多的启发和思考。对开源的内涵及定位有了一个较为清晰的认知。

- 在此过程中也掌握了一些基本的Github&Git的使用。(其实两年前就接触过Github，但仅仅是当作一个类似网盘的东西来用，根本不了解其有如此多的功能，也完全没意识到，那些大名鼎鼎的系统竟然都能在其上找到根源)。
- 参与的TiDB项目，由于精力有限，并没有真正提PR，关于数据一致性方面的问题，提出了Issue。通过阅读其相关文档，了解了其大概的工作原理及架构，因为自己之前调研过OLTP与OLAP混合系统的相关内容，也清楚的认识到TiDB是如何通过冷热数据区分里实现支持两种不同应用负载的。

5. 课程建议

这是一门特别新颖的课程，授课的方式也应当是新颖的。整体感觉较好，自己也学到了一些东西。但对整个课程的总体脉络把握的不清楚。开源的理念就是开放，共享，所以这门课也应该倾向于学生们一起共享、讨论，多增加一些学生的报告感悟等。同时，可以将课程分为几个大的模块，比如：技术专题(主要就是介绍像Github&Gitlab&Git的使用等)、项目专题(每周一个项目组的同学进行汇报展出)、开源分享会(老师或学生分享自己的在开源里的感悟，思考等)，几个专题可以并行亦可以串行。

Anyway,这门课我学到了很多知识，也认识到自己和当下最前沿的差距，在跌宕起伏的洪流中，我会继续努力最好，将来尽努力为开源贡献自己的微薄力量。