

《开源软件设计与开发》课程总结

姓名：邓思佳

学号：51194501112

1、开源理解：

开源的本质，是自由而不是免费。说到开源必然绕不过 Git 和 GitHub。下面我想谈谈对 Git 和 GitHub 的理解和实践。

Git 和 GitHub 区别和联系：

一些初次接触 Git 和 GitHub 的人常常将 Git 和 GitHub 二者混淆而谈。二者虽然联系甚紧，但从本质上是两个不同的概念。Git 是一个开源的分布式版本控制系统。而 GitHub 本质上是一个代码托管平台，它提供的是基于 Git 的代码托管服务。对于一个团队来说，即使不使用 GitHub，他们也可以通过自己搭建和管理 Git 服务器来进行代码库的管理，甚至还有一些其它的代码托管商可供选择，如 GitLab 等。值得一提的是 Git 作为一个开源项目，其代码本身就被托管在 GitHub 上，如果感兴趣，可以在 GitHub 上去一观其真容。Git 项目地址：<https://github.com/git>

Git：

在现代软件项目中，版本控制系统是项目配置管理不可或缺的一部分，甚至是其核心工具。早期的版本控制系统大多是集中式版本控制系统。业界主流的集中式版本控制系统包括 CVS、SVN 等。集中式版本控制很大程度上解决了版本控制和协同开发的问题，但是它也有重大的缺点。如果中央服务器出现宕机，那么开发者将无法提交代码，也无法进行协同工作，更无法查看文件历史。因此，目前业界最流行的版本控制系统是分布式版本控制系统，其最大的特点是各开发者本地所复制的不仅仅是当前最新版本的文件，而是把代码仓库完整地克隆了下来。

Git 是业界目前最为流行的分布式版本控制系统，可以做到号称"克隆一次代码库本地就拥有了一个完整的代码库副本，这听起来有些骇人听闻。

其实我们可以尝试从以下两个方面来理解：

其一，在分布式版本控制系统中，克隆了代码库的各本地开发者拥有了服务器分发过来的完整的代码库副本，使得开发者们可以独立于主服务器之外进行开发任务，这和分布式计算概念中，各计算机独立进行计算任务的理念不谋而合。同时也符合分布式存储的理念：一个文件多份副本。

其二，各开发者在完成开发任务后又需要将自己本地修改后的代码库合并（Merge）到主服务器上。这也与分布式计算概念中最终需要将各计算机的计算结果合并起来的概念是相符的。

通过实践比如尝试通过 `git reset` 命令将代码库恢复到任意目标 commit 的状态，我对 Git 克隆了完整的代码库副本有更加直观的理解。

2、开源贡献

在我参与的开源项目——Tidb 中，我做的贡献为实现为 #12106 的 `BuiltinCastDecimalAsDurationSig` 的矢量化评估，根据基准，比以前快了大约 5 倍。

在经过 6 次提交后成功合并到 `pingcap: master` 中，附上具体链接：<https://github.com/pingcap/tidb/pull/13790>

dengsigua commented on 27 Nov

Contributor + 😊 ...

What problem does this PR solve?

implement vectorized evaluation for builtinCastDecimalAsDurationSig, for #12106

What is changed and how it works?


according to benchmark, about 5 times faster than before:

```
BenchmarkVectorizedBuiltinCastFunc/builtinCastDecimalAsDurationSig-VecBuiltinFunc-8
BenchmarkVectorizedBuiltinCastFunc/builtinCastDecimalAsDurationSig-NonVecBuiltinFunc-8
```

Check List

Tests

- Unit test\


 francis0407 reviewed 28 days ago

View changes

francis0407 left a comment

Contributor + 😊 ...

LGTM

 qw4990 approved these changes 28 days ago

View changes

qw4990 left a comment

Contributor + 😊 ...

LGTM

3、课程反馈

转眼课程已经走向尾声了，在为即将结束一门课程而高兴的同时，更多的却是不舍和依恋。在说再见之前，我想还是应该留下点什么，在这里，谨提出我对这门课程的一些意见和建议，希望能够对老师以后的教学安排有所帮助。

其实有很多建议以前大家都已经提过了，比如说课程内容方面多注重讲技术啊等等。虽然这些建议不一定合适，但最起码这是大家对教技充满兴趣与热情的体现。除了这些，根据我自己的切身体会，还想再补充几点建议：

1. 我认为课程安排可以适时调整，比如一开始可以让最后一节课的学长现身说法，用自己的实际经历来讲述开源与我们息息相关，让我们意识到开源的价值与意义。
2. 我认为在一开始选开源项目之前可以将所有的可选项目的简介在课前以电子稿或纸质稿的形式发给大家，让大家先宏观的认识一下所有项目大致的要求和亮点。然后项目展示的时候大家就可以重点关注自己感兴趣的地

方，这样可能对项目理解的更透彻。

3. 在每个开源项目小组内部可以安排一个指导员，指导大家入门，或者简单介绍一下我们该怎么做。
4. 如果课时允许，可以抽几节课让同一个开源项目小组内部的进行答疑交流。

4、参考文献

[1]. Correction: Ten Simple Rules for Taking Advantage of Git and GitHub.[J]. PLoS computational biology,2019,15(6).

[2]倪光南. 开源软件在中国的发展[J]. 程序员, 2007(01):35.

[3]曾进群. 开源社区结构与行为及其特点研究[D]. 华南理工大学,2013.

[4]朱守园,王婷.基于 GIT 的软件开发模式探究[J].信息通信,2017(03):117-118.