

《开源软件设计与开发》课程总结

姓名：缪影芝

学号：51195100013

1、开源理解

最开始接触 git 是本科学 java 的时候，那个时候注册了自己的 github 账号，从此打开了自己知识的大门。在 github 上，人人都可以分享自己的代码，也可以下载别人的代码，在这上面，你可以看到很多的大牛，可以向很多优秀的人学习。

在开源课程之前我对于 github 的理解，仅仅停留在查资料、学习的阶段，通过学习别人的代码，或者别人的总结，可以让我收获很多，同时也可以激发我对于计算机这个领域的兴趣。那个时候的我对于开源并没有一个很清晰的概念，也没有思考过，为什么会有这样一个平台的出现，以及它存在的更深层次意义。

通过这本课程的学习，让我对开源有了新的认识。

在像 github 这样的开源平台上，每天都会有几百个开源项目加入、每天都有几十万的用户访问和下载开源软件，这反映了开源运动的蓬勃发展。开源本身是简单的，它无非是公开了源代码的计算机程序而已，它在电子计算机出现的同时就已经出现了。但是，开源软件运动的内涵却是博大精深的。

开放源代码开发模式与传统的开发模型的最显著区别是其非营利性和志愿参与性。开放源代码组织鼓励知识的共享而不是知识产权的保护，因此无法从研究开发的产品中获得利润，显然也就无法给与参与者物质上的激励。但是为何还是有很多人参与到开源中来，去贡献自己的一份力呢？这其中有很多因素。

在传统的企业中，虽然劳动者会获得工资，但是由于公司必须控制开发进度，往往给予员工既定的工作任务，而不是由员工自己选择感兴趣的项目。这样一来，员工感觉到自己是一部大机器中的一颗小螺丝，无法左右形势。在开源平台上，参与者可以选择自己的开发进度，选择自己的开发方向，通过研究别人的代码和参与讨论提高自己的能力。人类热爱自由的天性使开放源代码的这一特征使参与者得到了正常工资无法得到的自由的快乐。

并且在开源的环境下，当所有的程序员都可以免费获得某个程序的各种编码时，毫无疑问能够对程序进行改进，因为通过集体合作的力量可以找出原先程序中的各种错误加以改进或者可以因地制宜，开发出各种适应于不同需求和不同硬件平台的应用软件。

开源还可以大大提高创新能力，它提供了一种参与创造的非常低的进入壁垒。不论你是经验丰富的编程高手，还是在校大学生，你都可以参与到其中。牛顿之伟大是因为他“站在了巨人的肩膀上”，这是一种动态的创新而不是静态的所有权。如果对知识产权的保护变成了对创造力的制约，那么今天的牛顿可能要支付大量的许可费才能看到巨人高大的肩膀，更不用说“站在巨人的肩膀上”了。而开源为我们提供了这样一种平台，让参与者不用任何额外投资就提高自己的能力。

开源无论是对个人的学习还是对软件的发展，都有很大的积极作用。

在此之前，我从来没有自己参与过一个开源项目仅仅来停留在看别人的代码阶段，在本门课程中，通过参与一个开源项目，我了解了开源开发的一些详细过程。一个大的项目是非常耗时耗力的，但通过开源的方式，让更多人参与

到这个开发中来，可以大大提高效率，并且当我们自己参与其中的时候，可以通过与其他人的讨论与交流，发现自己代码中的问题，提高自己编程的能力。

2、开源贡献

这次我参与的是 tidb 项目，在参与过程中对 tidb 的项目结构有了一个了解，在这个过程中我首先 fork 了 tidb 的项目至自己的 github 仓库中，并完成了其中一个功能的开发，但是因为开发过程中有其他人先 merge 的该部分的代码，所以这部分代码提交的 pr 并没能成功 merge，但在这里我仍然简单介绍一下自己做了的工作。


我主要参与了 tidb 项目向量化工作中，尝试实现其中一个向量化的工作，具体代码如下：


```
▼ 28 expression/builtin_time_vec.go
@@ -716,11 +716,35 @@ func (b *builtinGetFormatSig) getFormat(format, location string) string {
716 716     }
717 717 }
718 718 func (b *builtinLastDaySig) vectorized() bool {
719 - return false
719 + return true
720 720 }
721 721
722 722 func (b *builtinLastDaySig) vecEvalTime(input *chunk.Chunk, result *chunk.Column) error {
723 - return errors.Errorf("not implemented")
723 + n := input.NumRows()
724 + buf, err := b.bufAllocator.get(types.ETDatetime, n)
725 + if err != nil {
726 +     return err
727 + }
728 + defer b.bufAllocator.put(buf)
729 + ds := buf.Times()
730 + result.ResizeTime(n, false)
731 + times := result.Times()
732 + for i := 0; i < n; i++ {
733 +     tm := ds[i].Time
734 +     year, month, day := tm.Year(), tm.Month(), tm.Day()
735 +     if month == 0 || day == 0 {
736 +         return handleInvalidTimeError(b.ctx, types.ErrIncorrectDatetimeValue.GenWithStackByArgs(buf.GetString(i)))
737 +     }
738 +     lastDay := types.GetLastDay(year, month)
739 +     ret := types.Time{
740 +         Time: types.FromDate(year, month, lastDay, 0, 0, 0, 0),
741 +         Type: mysql.TypeDate,
742 +         Fsp: types.DefaultFsp,
743 +     }
744 +     times[i] = ret
745 + }
746 +
747 + return nil
724 748     }
```

之后我对该代码提交了 pr。


expression: implement vectorized evaluation for 'builtinLastDaySig' #13592

 Closed myz000 wants to merge 1 commit into [pingcap:master](#) from [myz000:master](#) 

 Conversation **7**

 Commits **1**

 Checks **0**

 Files changed **1**



myz000 commented on 19 Nov

+ 😊 ...

What problem does this PR solve?

expression: implement vectorized evaluation for 'builtinLastDaySig'

What is changed and how it works?

goos: windows

goarch: amd64

pkg: github.com/pingcap/tidb/expression

BenchmarkVectorizedBuiltinTimeFuncGenerated-2 1000000000 0.0610 ns/op 0 B/op 0 allocs/op

BenchmarkVectorizedBuiltinTimeFunc-2 1000000000 0.0350 ns/op 0 B/op 0 allocs/op



PASS

Check List

Tests

- Unit test

项目 Reviewer 对我的代码进行了 review，并提出了修改意见，截图如下：

 wshwsh12 reviewed on 19 Nov [View changes](#)


wshwsh12 left a comment Contributor +😊 ...


Thanks for your contribution.
Please add some test cases to check correctness.

expression/builtin_time_vec.go

```
724 + buf, err := b.bufAllocator.get(types.ETDatetime, n)
725 + if err != nil {
726 +     return err
727 + }
728 + defer b.bufAllocator.put(buf)
```

Comment on lines +724 to +728


 wshwsh12 on 19 Nov Contributor +😊 ...
1. We can use result to get the son's result directly without alloc a buffer.
2. We need to call b.arg[0].VecEvalTime to get son's result.




[Resolve conversation](#)


expression/builtin_time_vec.go

```
730 + result.ResizeTime(n, false)
731 + times := result.Times()
732 + for i := 0; i < n; i++ {
733 +     tm := ds[i].Time
```

 wshwsh12 on 19 Nov Contributor +😊 ...
We need check Null before calculate.



但是最后由于有其他参与者告知他已经提交过该功能的 pr，所以我最后关闭了该 pr。

 @ekalinin commented Nov 20, 2019 Contributor +😊 ...

I'm sorry, but this function is vectorized by [#13503](#)

 myz000 closed this Nov 20, 2019

3、课程反馈

该门课程邀请了很多优秀的老师来进行讲解，开拓了我们的眼界，使我们对开源有了深入的认识。但是在学习的过程中，由于最开始很多同学对开源并不是很了解，或者没有使用过 git 等工具，那么他们就会对这门课程比较疑惑，并不知道老师在讲解什么，这个问题一般存在于刚开始学习的阶段，建议老师们在刚开始讲解的时候可以用更简单或者更容易激发同学们兴趣的例子，来调动同学们的积极性。

4、参考文献

- [1]陈伟. 开源社区在中国[J]. 软件世界, 2006(14):34-37.
- [2]陈晓伟. 开放源代码组织的激励机制及成功要素[J]. 特区经济, 2007(02):284-286.
- [3]王飞绒, 陈劲. 网络环境下新型的创新组织形式:开放源代码社区[J]. 研究与发展管理, 2006(06):37-42.