

《开源软件设计与开发》课程总结

姓名：匡博

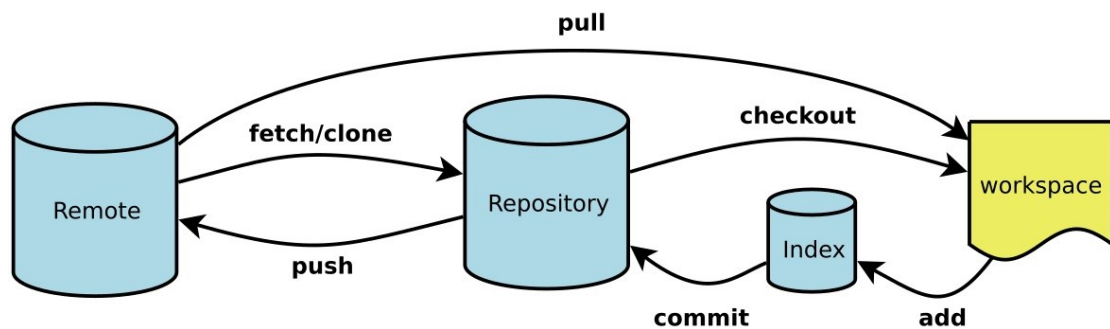
学号：51194507007

1、开源理解

在这之前，我对 GitHub 的理解仅仅停留在 GitHub 是一个纯粹的存储仓库，Git 工具也仅仅是下面这几行命令。

```
git config --global user.name "yourname"
git config --global user.email "youremail"
git init // 初始化仓库
git add 文件名 // 将文件添加到仓库
git commit -m "本次提交内容备注" // 提交
git status // 查看仓库当前的状态
git diff // 查看和 git 版本库有什么不同，查看修改了什么内容
git push origin master // 推送最新修改
git log // 打印提交的 log 日志
git clone https://github.com/KuangBo/final-assignment-open-source.git // clone 仓库
```

上面就是我入门阶段最先敲的几个命令了。本科期间也曾用过 GitHub，熟悉的也只是上面这几行命令，一直都认为 GitHub 是用 Git 的几行命令来回传文件，从来都不会去关心这些命令背后有什么，更是没有涉及到开源这个概念。



在之后的学习中慢慢地明白，Git 也不只是 Git，如上图所示，每一个都是一个独立的仓库。

本学期课程学习的过程，对我影响最大的可能就是开源了。一开始选这节课《开源软件设计与开发》的时候，单纯认为这门课可能就是教一教如何做软件开发，怎么进行软件架构的设计，然后怎么使用 GitHub。上了几节课发现大错特错，原来这节课的重点是在开源！不得不说，如果说真的是按照我一开始的想，这门课在教一些简简单单的软件开发的技能，没有一点价值。这门课程完完全全颠覆了我之前对教学的概念，让我切身地参与到了开源项目当中。

首先要说的就是参与过的 TiDB 项目了。在没有参与 TiDB 项目之前，对开源是那种既心动又感觉自己无法行动的那种感觉，感觉开源距离自己真的很遥远。在开了第一个 PR 之后，那种感觉无法言表，之前一直在使用 Apache Tomcat 与 Apache Struts 的时候，仅仅是将自己放在一个使用者的位置，而现在自己居然可以参与到 TiDB 开源项目当中。最令

人振奋的是，我居然非常幸运地参加了 2019 中国开源年会（COSCon'19），在现场见到了开源界很多厉害的人物。

在整个课程之中，感触最深的可能是最后一堂课前来讲课的赵生字前辈了，清本北硕，同济博士。因为 15 年埋下了一颗开源的种子，几年后生根发芽，前前后后有了这些成就。看生源，我同他肯定是无法相比的。在我硕士毕业之后能进得了他简历中的任意一家公司那简直就是不得了的事情啊。但是，我认为做开源事业或许并不看重出身，需要的是那一腔热血。不得不承认的是，这门课程确实实实在在我这埋下了一颗“不知名”的种子。待某一天应该也会生根发芽吧。

2、开源贡献

本学期参与的是 TiDB 项目，在参与的过程中，对如何开 PR、怎么去了解一个开源项目并参与进去等有了个大概的了解。我参加的是向量化工作 `Implement vectorized evaluation for 'builtinDecimalAnyValueSig'`，并对测试函数做了一些。提交的 PR 链接地址 <https://github.com/pingcap/tidb/pull/13159>。

```
expression/builtin_miscellaneous_vec.go
@@ -159,11 +159,11 @@ func (b *builtinNameConstStringSig) vecEvalString(input *chunk.Chunk, result *ch
159 159     }
160 160
161 161     func (b *builtinDecimalAnyValueSig) vectorized() bool {
162 162         - return false
162 162         + return true
163 163     }
164 164
165 165     func (b *builtinDecimalAnyValueSig) vecEvalDecimal(input *chunk.Chunk, result *chunk.Column) error {
166 166         - return errors.Errorf("not implemented")
166 166         + return b.args[0].VecEvalDecimal(b.ctx, input, result)
167 167     }
168 168
169 169     func (b *builtinUUIDSig) vectorized() bool {

expression/builtin_miscellaneous_vec_test.go
@@ -53,6 +53,7 @@ var vecBuiltinMiscellaneousCases = map[string][]vecExprBenchCase{
53 53     ast.AnyValue: {
54 54         {retEvalType: types.ETDuration, childrenTypes: []types.EvalType{types.ETDuration}},
55 55         {retEvalType: types.ETInt, childrenTypes: []types.EvalType{types.ETInt}},
56 56         + {retEvalType: types.ETDecimal, childrenTypes: []types.EvalType{types.ETDecimal}},
56 57         {retEvalType: types.ETTimestamp, childrenTypes: []types.EvalType{types.ETTimestamp}},
57 58         {retEvalType: types.ETReal, childrenTypes: []types.EvalType{types.ETReal}},
58 59         {retEvalType: types.ETString, childrenTypes: []types.EvalType{types.ETString}},
60 60     },
61 61 },
62 62
63 63     @@ -68,18 +69,18 @@ var vecBuiltinMiscellaneousCases = map[string][]vecExprBenchCase{
68 69     },
69 70     }
70 71
71 71     - func (s *testEvaluatorSuite) TestVectorizedBuiltinMiscellaneousCasesEvalOneVec(c *C) {
72 72     + func (s *testEvaluatorSuite) TestVectorizedBuiltinMiscellaneousEvalOneVec(c *C) {
72 73         testVectorizedEvalOneVec(c, vecBuiltinMiscellaneousCases)
73 74     }
74 75
75 75     - func (s *testEvaluatorSuite) TestVectorizedBuiltinMiscellaneousCasesFunc(c *C) {
76 76     + func (s *testEvaluatorSuite) TestVectorizedBuiltinMiscellaneousFunc(c *C) {
76 77         testVectorizedBuiltinFunc(c, vecBuiltinMiscellaneousCases)
77 78     }
78 79
79 79     - func BenchmarkVectorizedBuiltinMiscellaneousCasesEvalOneVec(b *testing.B) {
80 80     + func BenchmarkVectorizedBuiltinMiscellaneousEvalOneVec(b *testing.B) {
80 81         benchmarkVectorizedEvalOneVec(b, vecBuiltinMiscellaneousCases)
81 82     }
82 83
83 83     - func BenchmarkVectorizedBuiltinMiscellaneousCasesFunc(b *testing.B) {
84 84     + func BenchmarkVectorizedBuiltinMiscellaneousFunc(b *testing.B) {
84 85         benchmarkVectorizedBuiltinFunc(b, vecBuiltinMiscellaneousCases)
85 86     }
```

以上是对`builtinDecimalAnyValueSig`向量化的代码，以及对测试函数修改之后的代码。下面是对该代码提交 PR 并进行讨论。由于是第一次提交，而且没有什么经验，该 PR 历经五天，提交了 18 次，终于 Merged。在提交过程中，经过 Reviewer 耐心的指导，成功修改了测试函数。

通过使用提交 PR 的模板，学到在提交 PR 之前应该仔细描述该 PR 解决要解决什么问题，它改变了什么，运行起来怎么样，经过压力测试的性能怎么样，所进行的测试又是什么测试。例如 Implement vectorized evaluation for `builtinDecimalAnyValueSig` 这个 PR 的提交。

expression: implement vectorized evaluation for `builtinDecimalAnyValueSig` #13159

Merged qw4990 merged 18 commits into pingcap:master from KuangBo:master on 9 Nov

Conversation 23 Commits 18 Checks 0 Files changed 2



KuangBo commented on 5 Nov • edited

Contributor + 😊 ...

What problem does this PR solve?

expression: implement vectorized evaluation for builtinDecimalAnyValueSig

What is changed and how it works?

```
go test -v -benchmem -bench=BenchmarkVectorizedBuiltinMiscellaneousFunc -run=BenchmarkVectorizedBuiltinM
goos: linux
goarch: amd64
pkg: github.com/pingcap/tidb/expression
BenchmarkVectorizedBuiltinMiscellaneousFunc/builtinDecimalAnyValueSig-VecBuiltinFunc-4      755986
BenchmarkVectorizedBuiltinMiscellaneousFunc/builtinDecimalAnyValueSig-NonVecBuiltinFunc-4    53552
PASS
ok github.com/pingcap/tidb/expression 2.545s
```

Check List

Tests

- Unit test



SunRunAway commented on 6 Nov • edited ▾

Member



The name of test functions and benchmark functions in builtin_miscellaneous_vec_test.go is not well formatted. Please help to correct it.

```
TestVectorizedBuiltinMiscellaneousCasesEvalOneVec ->
```

```
TestVectorizedBuiltinMiscellaneousEvalOneVec
```

```
TestVectorizedBuiltinMiscellaneousCasesFunc -> TestVectorizedBuiltinMiscellaneousFunc
```

```
BenchmarkVectorizedBuiltinMiscellaneousCasesEvalOneVec ->
```

```
BenchmarkVectorizedBuiltinMiscellaneousEvalOneVec
```

```
BenchmarkVectorizedBuiltinMiscellaneousCasesFunc ->
```

```
BenchmarkVectorizedBuiltinMiscellaneousFunc
```

After that, you can try `make vectorized-bench VB_FILE=Miscellaneous VB_FUNC=builtinDecimalAnyValueSig`



KuangBo commented on 6 Nov

Author

Contributor



@SunRunAway Thanks. My results are as follows:

```
go test -v -benchmem -bench=BenchmarkVectorizedBuiltinMiscellaneousFunc -run=Bench
goos: linux
goarch: amd64
pkg: github.com/pingcap/tidb/expression
BenchmarkVectorizedBuiltinMiscellaneousFunc/builtinDecimalAnyValueSig-VecBuiltinFunc
BenchmarkVectorizedBuiltinMiscellaneousFunc/builtinDecimalAnyValueSig-NonVecBuiltin
PASS
ok  github.com/pingcap/tidb/expression  2.545s
```

```
go test -v -benchmem -bench=BenchmarkVectorizedBuiltinMiscellaneousCasesFunc -run=E
goos: linux
goarch: amd64
pkg: github.com/pingcap/tidb/expression
BenchmarkVectorizedBuiltinMiscellaneousCasesFunc/builtinDecimalAnyValueSig-VecBuiti
BenchmarkVectorizedBuiltinMiscellaneousCasesFunc/builtinDecimalAnyValueSig-NonVecE
PASS
ok  github.com/pingcap/tidb/expression  3.600s
```



SunRunAway commented on 7 Nov

Member



Hi, @KuangBo

You can post your benchmark result into the PR description, Thanks.



KuangBo commented on 7 Nov

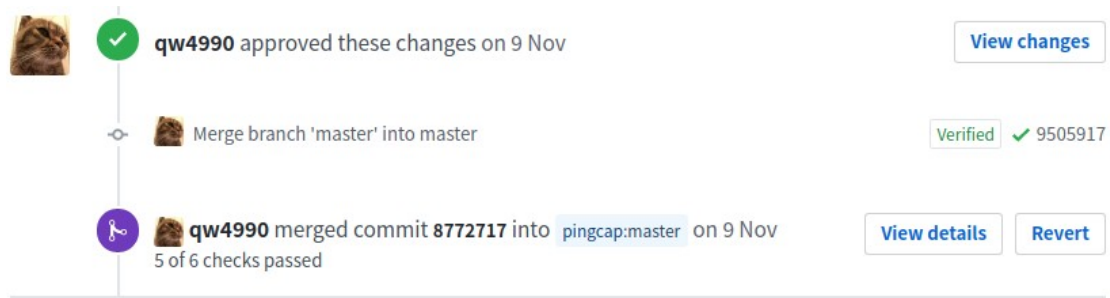
Author

Contributor



Oh, thank you @SunRunAway

This is my negligence, I have modified it.



A snippet of a Git commit history. The top entry shows a green checkmark icon, a profile picture, and the text "qw4990 approved these changes on 9 Nov". To the right is a button labeled "View changes". Below this is a merge commit entry with a key icon, a profile picture, and the text "Merge branch 'master' into master". To the right is a green box labeled "Verified" and a green checkmark followed by "9505917". The bottom entry shows a purple merge icon, a profile picture, and the text "qw4990 merged commit 8772717 into pingcap:master on 9 Nov". Below this text is "5 of 6 checks passed". To the right are two buttons: "View details" and "Revert".

3、课程反馈

还是之前的那句话，这门课程颠覆了我对教学的定义。之前我一直都认为教学就是老师在讲台讲课，学生在下面听课，老师们也确实都在这么做，虽然大家一直都很清楚教学的定义是教与学。该课程邀请了很多优秀的老师或者说是企业讲课，开拓了我们的眼界，让我们参与其中，我们在里面学到很多。所以说我认为该课程一开始的定位就是正确的，大学的课程应该就是多一些这类教学方法的课程，而不再是简单地念 PPT，让同学们参与到行业前列中。

但是，同学们的配合度并不高，也有很多同学并没有真正地参与到这门课程中。建议老师可以在课程安排中多一些分组讨论或者分组报告的形式，让同学们参与到开源课程中。

4、参考文献

[1]Scott Chacon,Ben Straub.Pro Git[EB/OL].<https://git-scm.com/book/en/v2>,2014-11-9.

[2]廖雪峰.Git 教程[EB/OL].<https://www.liaoxuefeng.com/wiki/896043488029600>.

[3]慕课网.Git 教程[EB/OL].<https://www.zhihu.com/question/38008771>.