



一只大尾巴狼的博客

对方不想和你说话，并向你扔了一条狗...

博客园

首页

新随笔

联系

订阅

管理

随笔 - 12 文章 - 0 评论 - 19 阅读 - 40860

Git如何fork别人的仓库并作为贡献者提交代码

例如 要fork一份google的MLperf/inference代码，下面介绍具体做法：

预备知识

git里的参考有几种表示，分别是上游仓库，远程仓库和本地仓库，逻辑关系如下

拉取代码的顺序：

别的大牛的代码(上游仓库)----->你fork的代码(远程仓库)----->你电脑的代码(本地仓库)

提交代码的顺序：

别的大牛的代码(上游仓库)<-----你fork的代码(远程仓库)<-----你电脑的代码(本地仓库)

每个仓库主分支是master，还可以有其它分支

上游仓库的表示为 upstream，远程仓库表示为origin

具体步骤

1、进入MLperf/inference仓库，点击fork按钮，拷贝代码到自己的git账号的仓库里



2、之后再自己的账号里就可以看到fork的代码了，可以看到代码的fork来源，即上游仓库地址



3、下载代码到你的本地电脑里，windows可以使用GitBash工具类似linux命令行的操作

下面是在本地电脑里进行的操作：

```
# 创建inference目录
$ mkdir inference
# 切换到inference目录
$ cd inference
# 创建并初始化git库
$ git init
# 添加远程git仓库
$ git remote add origin https://github.com/yananYangYSU/inference.git
# 添加SSH秘钥到git远程库，邮箱可以从git账号里查看
$ ssh-keygen -t rsa -C "1131074225@qq.com"
```

公告

昵称： 一只大尾巴狼
园龄： 4年1个月
粉丝： 5
关注： 5
[+加关注](#)

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

linux(3)
云服务(2)
虚拟化(2)
docker(2)
内核编译(1)
riscv(1)
L2TP(1)
VPN代理(1)
分区存储(1)
查询优化(1)
更多

随笔档案

2019年11月(2)
2017年10月(3)
2017年9月(7)

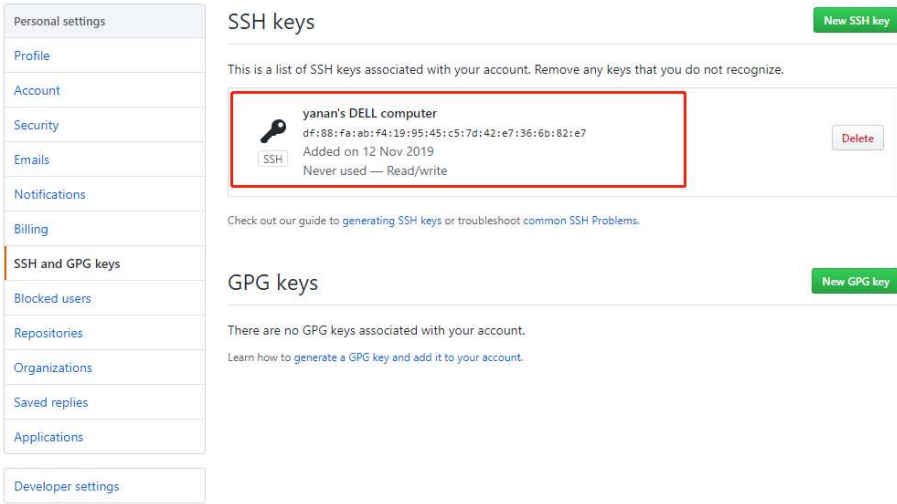
阅读排行榜

```
$ ssh-keygen -t rsa -C "1131074225@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/DELL/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/DELL/.ssh/id_rsa.
Your public key has been saved in /c/Users/DELL/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:GPS05a3b9iSKEVxjZayMnwfvUoYvJq1xk+su4oZaTbg 1131074225@qq.com
The key's randomart image is:
+---[RSA 3072]-----+
|
| . . . . o
| . 0 + + .
| . 00 = . .
| . + . 0 + 0
| . . . S . =
| + . . 0 + 0 =
| E . . . = . * + .
| .. 0 . 00 = + 0 +
| .. 0 . . . = 0 . 0 .
+---[SHA256]-----+
```

```
# 查看秘钥
cat ~/.ssh/id_rsa.pub
```

```
DELL@DESKTOP-GDL2BNH MINGW64 /d/gitWorkspace/inference (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAAB3NzaC1yc2EAAAADAQABAAQGDhstZ16Yi0cz1QTrc/+F4yAhwYHmtVT1sI/En2TJx2
GlnXfDdDr/5QpkZctJAUz...ctyQLHXX4DhxasaGjpunGr
IeaM3LQ06bu8bm8oXsdLvJ...m2r7xxrhIP7ytGtEtLLk9djmSt
9r6gJ98m/2aJUb2iDqA0...VDFMONZnaYB02haxX7LLmehCfntdByr38m6YFvvvH9hYIgY58rAIhPBck
wO947841DTARieLkFMq...r2D8czmThtJK3yg7EZcWdZ9
XINQ525fKcSDG7sMp3Mjh/OJ+eunwLp...uGf/KbRI dsPgwdTrxHTR8qH1
xBSSyyM03JNra1k3200nK2bKBH1PUsXcsmwUuv4PoSLPreS2touEFbvz6GaTYKSSU0y0js= 1131074
225@qq.com
```

复制添加到你git账号里的ssh key列表里，就可以通过安全认证传输数据了



```
# 将远程git库代码下载到本地 (origin代表远程仓库，master代表主分支)
git pull origin master
```

```
DELL@DESKTOP-GDL2BNH MINGW64 /d/gitWorkspace/inference (master)
$ git pull origin master
remote: Enumerating objects: 5097, done.
Receiving objects: 23% (1219/5097), 5.06 MiB | 29.00 KiB/s
```

代码下载完了之后，这时候你本地仓库和远程仓库的代码一致了，接下来对代码进行修改

4、修改本地仓库的代码并提交到远程仓库
假设我们要修改README.md文件，有两种方式可供选择：

```
-----
第一种方式，直接在master分支上做修改，命令如下：
# 修改README.md
$ echo "abc" >> README.md
# 添加要提交的临时文件
$ git add README.md
# 提交更改
$ git commit -m 'commit'
# 将本地仓库的修改上传到远程仓库
$ git push -u origin master
-----
```

1. MySQL千万级数据分区存储及查询优化 (21685)
2. qemu模拟器下编译运行基于riscv指令集的Linux操作系统(9646)
3. Git如何fork别人的仓库并作为贡献者提交代码(3255)
4. 博客园添加侧边栏小插件并更改css样式 (3188)
5. 基于Java的支持可变QPS的http负载生成器，提供交互界面和RMI接口(861)

评论排行榜

1. qemu模拟器下编译运行基于riscv指令集的Linux操作系统(9)
2. 博客园添加侧边栏小插件并更改css样式 (7)
3. 19组-第1次作业(2)
4. 19组-第二次作业(1)

推荐排行榜

1. Git如何fork别人的仓库并作为贡献者提交代码(4)
2. MySQL千万级数据分区存储及查询优化 (3)
3. 博客园添加侧边栏小插件并更改css样式 (3)
4. qemu模拟器下编译运行基于riscv指令集的Linux操作系统(2)
5. 19组-第1次作业(1)

最新评论

1. Re:博客园添加侧边栏小插件并更改css样式
@ 康先 正解...
--一只大尾巴狼
2. Re:博客园添加侧边栏小插件并更改css样式
我发现用这个在chrome内核下不能显示，ba把代码里的http改成https就可以了。不知道为什么，菜鸟一只。
--康先森
3. Re:qemu模拟器下编译运行基于riscv指令集的Linux操作系统
嗯，很高兴认识你，我现在在做riscv指令架构的一些程序移植实验，以后可以多多交流，方便加一个微信吗？我的微信号 YangYanan2333
--你的代码有问题
4. Re:qemu模拟器下编译运行基于riscv指令集的Linux操作系统
@ 你的代码有问题您好，我是对开源指令集感兴趣的开发人员，比较关注工程实践，很高兴能够与您交流，国内目前对RISCv的关注没有那么热，非常希望后续再您的博客中看到更多的相关内容。...
--hellmonky
5. Re:qemu模拟器下编译运行基于riscv指令集的Linux操作系统
@ hellmonky你好 你也在做riscv的研究，是做什么方向的啊，以后可以一起交流...
--你的代码有问题

第二种方式，在master分支基础上切出一个dev分支，然后在dev分支上修改，修改完成后，再将dev分支

merge到master分支

首先切换到dev分支

```
$ git checkout -b dev
```

```
$ echo "abc" >> README.md
```

添加要提交的临时文件

```
$ git add README.md
```

提交更改

```
$ git commit -m 'commit'
```

将本地仓库的修改上传到远程仓库

```
$ git checkout master
```

```
$ git merge dev
```

将本地仓库的修改上传到远程仓库

```
$ git push -u origin master
```

注意如果再push之前，上游仓库有人做了代码修改，那么此时你本地和远程仓库的代码就不是最新的了，此时需要把上游仓库的代码合并到本地，然后再push到远程仓库

添加上游仓库地址

```
$ git remote add upstream https://github.com/mlperf/inference.git
```

查看 origin 和 upstream 对应的仓库是否正确

```
$ git remote -v
```

```
$ git remote -v
origin  https://github.com/yananYangYSU/inference.git (fetch)
origin  https://github.com/yananYangYSU/inference.git (push)
upstream https://github.com/yananYangYSU/inference.git (fetch)
upstream https://github.com/yananYangYSU/inference.git (push)
```

origin对应的是自己github的地址，即yourname/project

upstream对应的是原项目的地址，即sourcename/project

从上游仓库获取最新的代码合并到自己本地仓库的master分支上

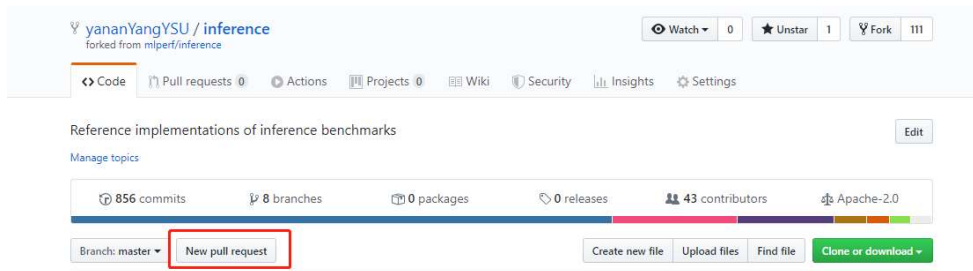
```
$ git pull upstream master
```

推荐每次代码待提交前，都从原项目拉取一下最新的代码，最后再使用git push命令将改动同步到自己的Github远程仓库中：

```
$ git push -u origin master
```

5、将远程仓库的代码提交到上游仓库：

进入GitHub账号的远程仓库，此时已经能够看到刚刚从本地仓库提交的修改了，然后点击New pull request



进入结果对比页面，如下图所示：

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



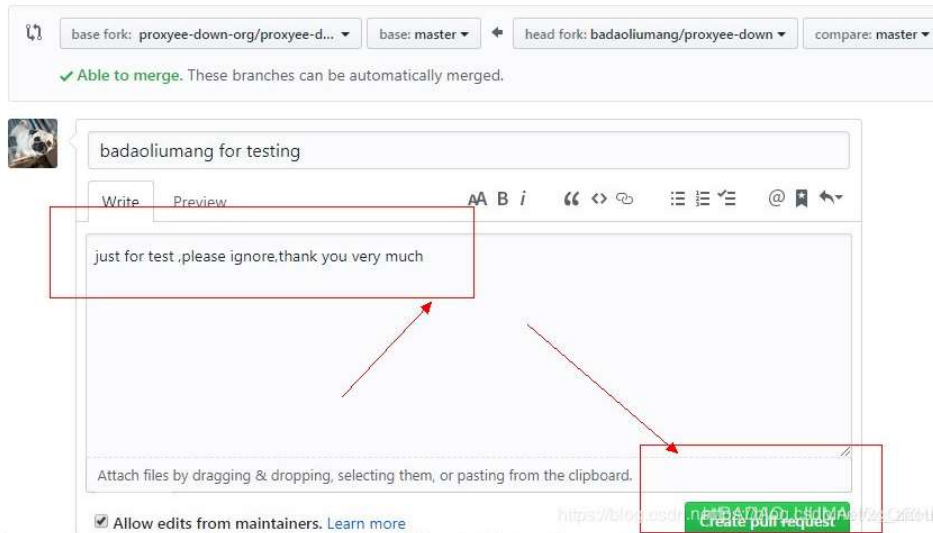
可以看到从远程库的master分支向上游仓库的master分支申请提交代码

Able to merge代表你的代码与上游代码没有冲突，可以提交

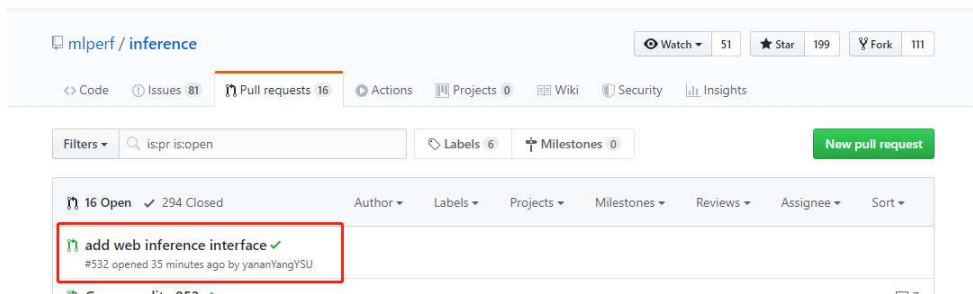
然后点击 create pull request，进入下面页面：

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



填写注释，描述你所作的修改，然后点击右下角提交

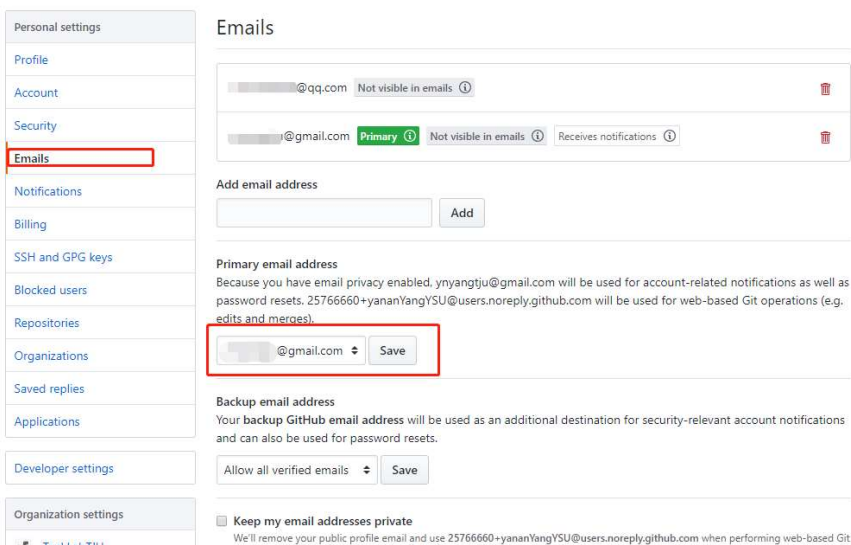


然后进入上游仓库的地址，在上游仓库的Pull requests列表里就可以看到自己的提交请求了，等待作者审核即可

附录：

如果是google的代码，那么提交代码要同意开发者协议，此时需要注册Gmail邮箱，申请Google Individual CLA，然后使用通过验证的gmail邮箱去修改并提交代码，google的robot才会通过你的代码提交验证，不然提交的pull request申请是无法通过校验的

准备好gmail邮箱后，在github的账号里配置个人配置github邮箱



然后在git命令行里也要设置提交代码的邮箱

```
git config --global user.email example@gmail.com
```

然后再进行步骤1

博主原创，转载请标明出处！谢谢

好文要顶

关注我

收藏该文

一只大尾巴狼

关注 - 5

粉丝 - 5

+加关注

« 上一篇：[19组第四次作业——使用四象限法分析本小组项目](#)

» 下一篇：[基于Java的支持可变QPS的http负载生成器，提供交互界面和RMI接口](#)

posted @ 2019-11-15 17:10 一只大尾巴狼 阅读(3255) 评论(0) 编辑 收藏 举报

40

刷新评论 刷新页面 返回顶部

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

- 编辑推荐：
- 在 ASP.NET Core Web API中使用 Polly 构建弹性容错的微服务
 - 带团队后的日常思考（五）
 - 聊聊我在微软外服的工作经历及一些个人见解
 - 死磕 NIO — Reactor 模式就一定意味着高性能吗？
 - 消息队列那么多，为什么建议深入了解下RabbitMQ？

- 最新新闻：
- 小米真无线降噪耳机 3 Pro 体验：应有所有的旗舰，还有需要打磨的细节（2021-10-27 21:30）
 - 新版 Photoshop 来了！悬停抠图、一键换季，还能和 AI 联动（2021-10-27 21:20）
 - 中日电池厂商，决战苹果造车供应链（2021-10-27 21:05）
 - 字节跳动确认已启动退出李子柒签约公司 “微念” 投资流程（2021-10-27 20:53）
 - 微软CEO纳德拉：大规模AI模型、人机界面或迎来大突破（2021-10-27 18:50）
- » 更多新闻...