如何在git仓库中彻底清除大文件

2016-07-17 • Git • Comments

> 坑的由来
>
> 事情是这样的，相信每个公司都有自己的官网。我们也不例外，而由于历史及各种不具名的原因，我司先前的程序猿哥哥将许多视频放在了代码仓库里。结果可想而知，原本只有代码的仓库突然间变得无比臃肿(或者是慢慢臃肿)，从早期的几十MB,迅速飙升至1G以上。眼看要挤爆服务器。这时我们总监看不下去了，大手一挥删了所有视频。仓库眼看瞬间缩小。但是，理论上是该缩小的，直到你不经意间瞄了眼——又多了一个G，这时你紧张了，慌了，明明删了视频，怎么仓库竟还变大了。
>
> 这一期，让我们走进git,细细体会git带给我们的欢乐与忧愁。

关于git

- 每个人都会有过去，git也不例外。
  git是个什么东西

到底发生了什么

早些时候我对git的原理并不是很了解，只是随着日常使用，终于开始理解git其实是一个指针指向一次提交的对象，当你在各个分支间切换的时候，指针就随之切换，版本也随之更改。

那么，git 是如何做到的能在各个版本间无缝切换的呢。即使long long ago的代码，只要来一句`git reset --hard sesd54f54sdf5sd4sd5f`照样给你打回原形。

真相只有一个，那就是其实所有的版本，不管是否存在了多久，都仍然存在于硬盘里。所以你才可以任性地对代码为所欲为。然而，为所欲为也要付出代价。

代价就是，你删了几次，就会有几个快照存在于硬盘里。删一个大视频，表面上少了500M空间，实则增加了一次至少500M的历史提交记录，虽然现在的代码仓库里不再有这个视频，但是你试试`du -sh .git`看看.git 文件大小，是不是有惊喜？我在这儿体会的最恐怖的一次，见证了3个多G的.git。

如何 `solve it`

- 首先我们看看git相关文件占用的空间，运行`git count-objects -v`

```
count: 35
size: 4404
```

```
in-pack: 20775
packs: 1
size-pack: 1502667
prune-packable: 0
garbage: 1
size-garbage: 136679
```

- `size-pack` 以千字节为单位表示，那么这里就有1.5G大小，这对代码仓库来说可是个恐怖的数字了。

- 那么让我们来找出罪魁祸首——到底是哪些大文件在混淆视听。
- 将所有含这些大文件的历史提交记录，一个不漏的找出来。

  - `git verify-pack` 可以识别出大对象，使用 `sort` 参数对输出的5列信息排序，再进行定向
  - 运行 `git verify-pack -v .git/objects/pack/pack-15esdkfdksjfd...asd.idx | sort -k 3 -n | tail -5`
  - 如果你的仓库确实有点臃肿的话，让代码飞一会儿。
  - 得到以下结果:

```
b9e5fc1ea2a87e95ddea6fed47f29c184595485b blob   70580712 62447447 1360078
b8b3833e7f8e8207033b9f7e34e497a3207ecae2 blob   103319718 102366768 11572
bc4f928a65692d19ab8d9378778463c13be7204c blob   117527265 117467079 10267
52565aeb9009bb4d07ca8d7c130425d9bf31f7b2 blob   201567615 181006235 78816
98c5dc977918aaf3b45a999ae92e031b4b15191b blob   590381017 589431074 19873
```

- **unbelievable!**都是huge的文件!最底下的那家伙，到底是什么啊！

- 让我们用rev-list命令来查看commit 的 SHA值和文件路径:
- `git rev-list --objects --all | grep 98c5dc9779`

```
98c5dc977918aaf3b45a999ae92e031b4b15191b public/video/feng-mv.mp4
```

- 这次倒很快，答案揭晓——是公司宣传视频。我们明明有七牛。这玩意儿就不该出现在这里。

- 再来看看所有包含这家伙的提交历史:
- `git log --pretty=oneline --branches -- public/video/feng.mp4`

```
9364a4f65f03d3a6d0873106ba21bd3172418176 chore: 删除视频文件
da24a01e46876a495518f9617501b4b360177f2b improve(官网首页)
```

- 一个大文件的一删一减两个操作，就会占据很多空间，更何况还有其他的较大文件。

- 找到了根源所在，是时候让我们大干一场:

  - `git filter-branch --index-filter 'git rm --ignore-unmatch --cached public/video/feng.mp4' -- da24a01^..`
  - 使用 `filter-branch` 重写所有相关的提交历史。

- `index-filter` 修改暂存区域或索引。
- `git rm --cache` 从git仓库里删除文件。
- `ignore-unmatch` 忽略不匹配文件的提交记录，不会抛出错误。
- 最后一个哈希值指重写从这次提交开始的所有提交记录，避免索引所有的提交历史
- 好了，开始运行吧。一阵眼花缭乱的滚动之后，它宣布重写完成。

```
...
Rewrite 38f2b11070b7e21cbce465b5f6b384156e3a8e61 (475/475)
Ref 'refs/heads/master' was rewritten
```

- 此时的历史提交记录中已经不再会有指向那位大文件的引用了。

- 但仍然需要以下两条命令来删除refs对他的引用(关于这部分的理解我也有待深究)。

  你的历史中将不再包含对那个文件的引用。 不过，你的引用日志和你在 .git/refs/original 通过 filter-branch 选项添加的新引用中还存有对这个文件的引用，所以你必须移除它们然后重新打包数据 库。 在重新打包前需要移除任何包含指向那些旧提交的指针的文件:

```

rm -Rf .git/refs/original
rm -Rf .git/logs/
git gc
Counting objects: 21406, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (9323/9323), done.
Writing objects: 100% (21406/21406), done.
Total 21406 (delta 11867), reused 20640 (delta 11348)
```

- 再看看节约了多少空间:

```
git count-objects -v
warning: garbage found: .git/objects/pack/tmp_pack_zG7GCb
count: 154
size: 4884
in-pack: 21406
packs: 1
size-pack: 1502752
prune-packable: 0
garbage: 1
size-garbage: 136679
```

- 尽管表面上看去没有什么软用，但是当我运行 `du -sh .git` 查看大小时，发现.git文件夹确实缩小了。

- 现在这个视频文件存在于松散对象(size)中。虽然它没有彻底消失，但已经不再出现于推送或克隆中。
- 如果需要彻底移除，运行 `git prune --expire now` 。
- 再次查看 `git count-objects -v` 。

小结

- 讲道理走到这一步应当是成功了，只是刚才运行`git count-objects -v`的结果并不那么尽如人意，它并没有缩减的很明显。
- 这点在以后解决成功时再另起一篇。
- 最后一点–作为一个前端攻城狮，可不能什么都往城墙里放啊!!!

#git

< ES6 简单特性尝鲜-1

webpack小试牛刀 >