Scrapyd 附加知识

底层框架

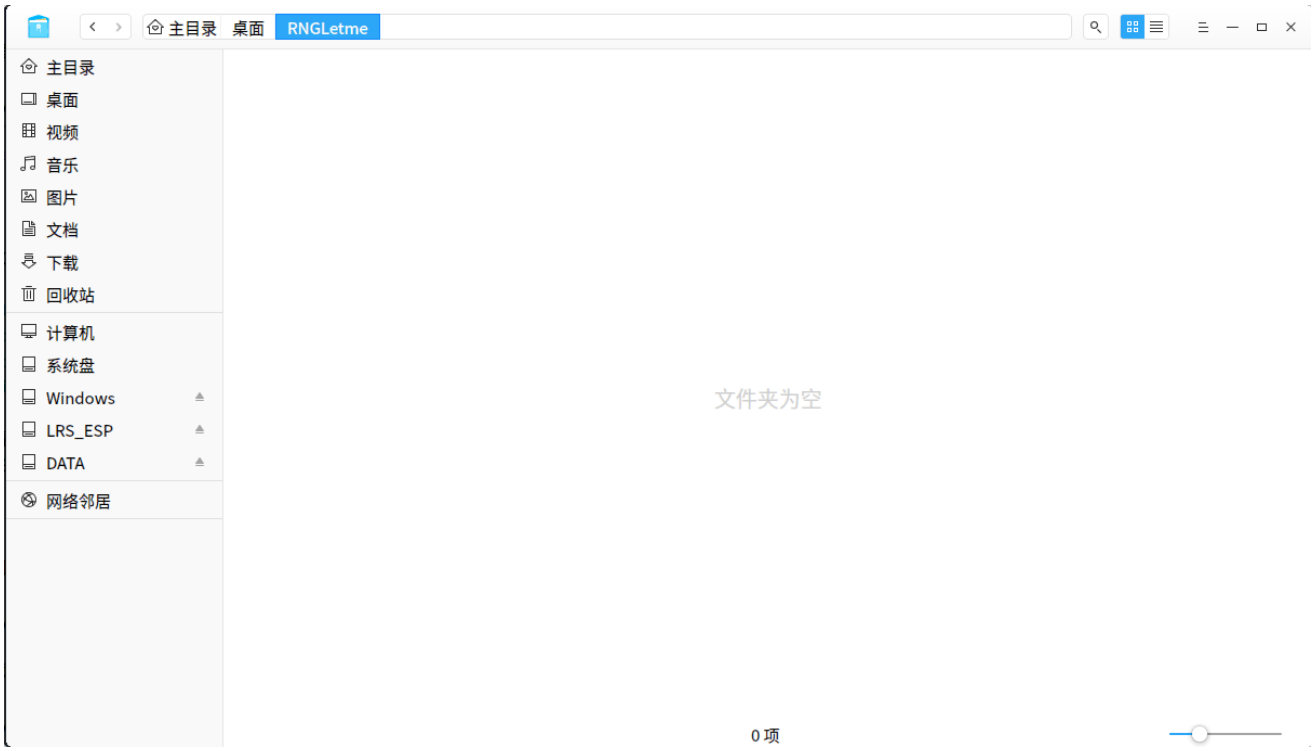
Scrapyd 与Scrapy 一样，底层框架都是 Twisted。不同的是 Scrapyd 较多的使用了 Twisted 的 Web 部分（当然了，Scrapyd-client 也一样）。

Eggs、Logs、Dbs 目录

配置文件中可以指定`eggs_dir`和`logs_dir`以及`dbs_dir`的路径：

```
eggs_dir    = eggs
logs_dir    = logs
dbs_dir     = dbs

```

默认情况下它们会在当前启动目录寻找是否有指定的同名文件夹，如果有则使用，否则根据配置新建文件夹。假设笔者在 RNGLetme 目录内启动项目，是否会在 RNGLetme 目录内生成对应的文件夹呢？



事实上，当在指定目录内启动 Scrapyd 时，会根据配置在当前目录生成`dbs`文件夹以及`twisted.pid`。那 eggs 和 logs 呢？

笔者猜它们肯定是在使用的时候生成，继续这个实验，在目录内新增一个 Scrapy 项目，并且将其打包部署到本机的 Scrapyd 服务，然后通过 API 启动项目中的爬虫，看看文件目录有什么变化：

动态图中演示了整个过程，可以看到 eggs 和 logs 在使用的时候才会生成，并非是项目启动就会生成。你也可以为其指定目录，如：

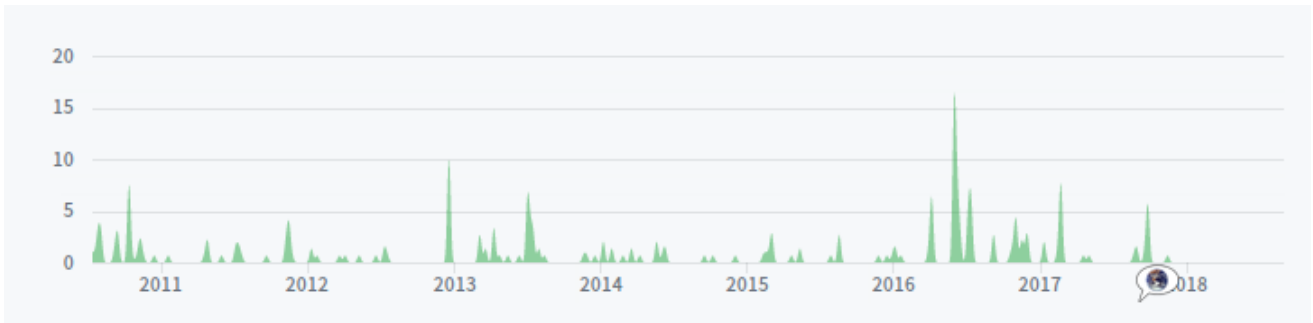
```
eggs_dir    = /home/rng/eggs
logs_dir    = /home/rng/logs
dbs_dir     = /home/rng/dbs

```

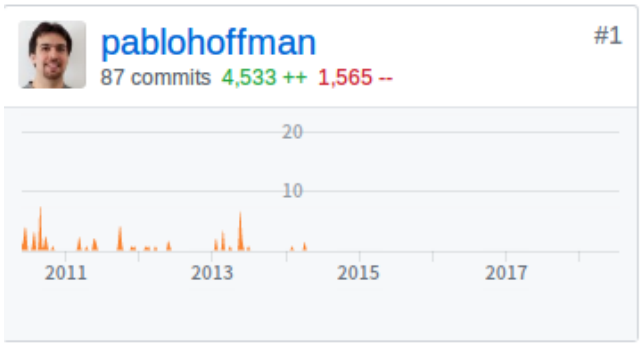
你可以尝试一下，看它们是否会乖乖的在指定目录下生成。

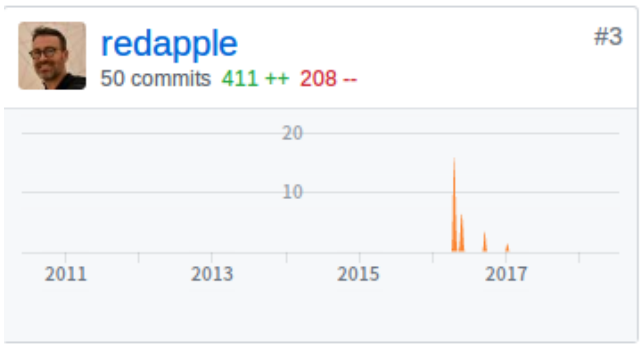
开发团队

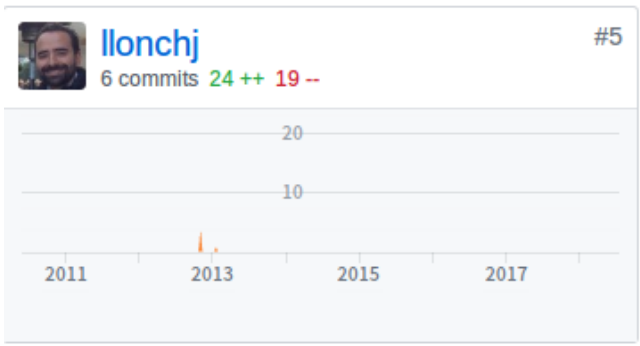
Scrapyd 与 Scrapy 是同一个开发团队负责的，先有 Scrapy，后有 Scrapyd。

你可以在 GitHub 上看到每个开发者的贡献量:开发者名单

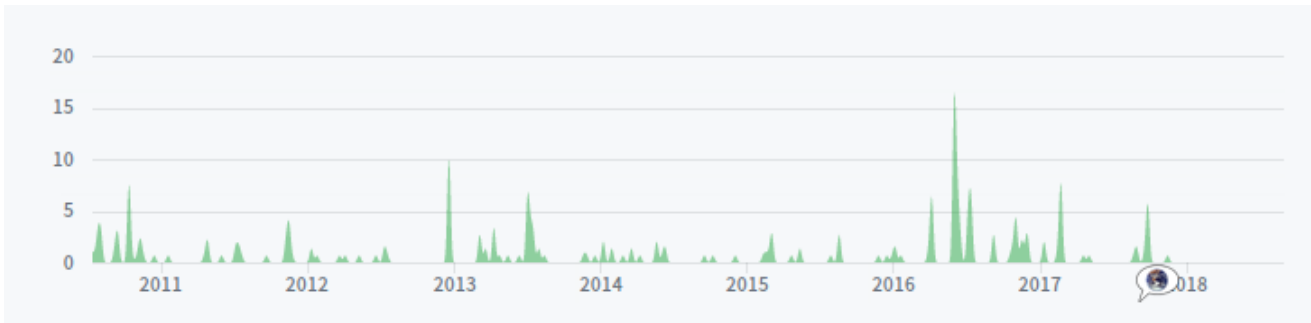








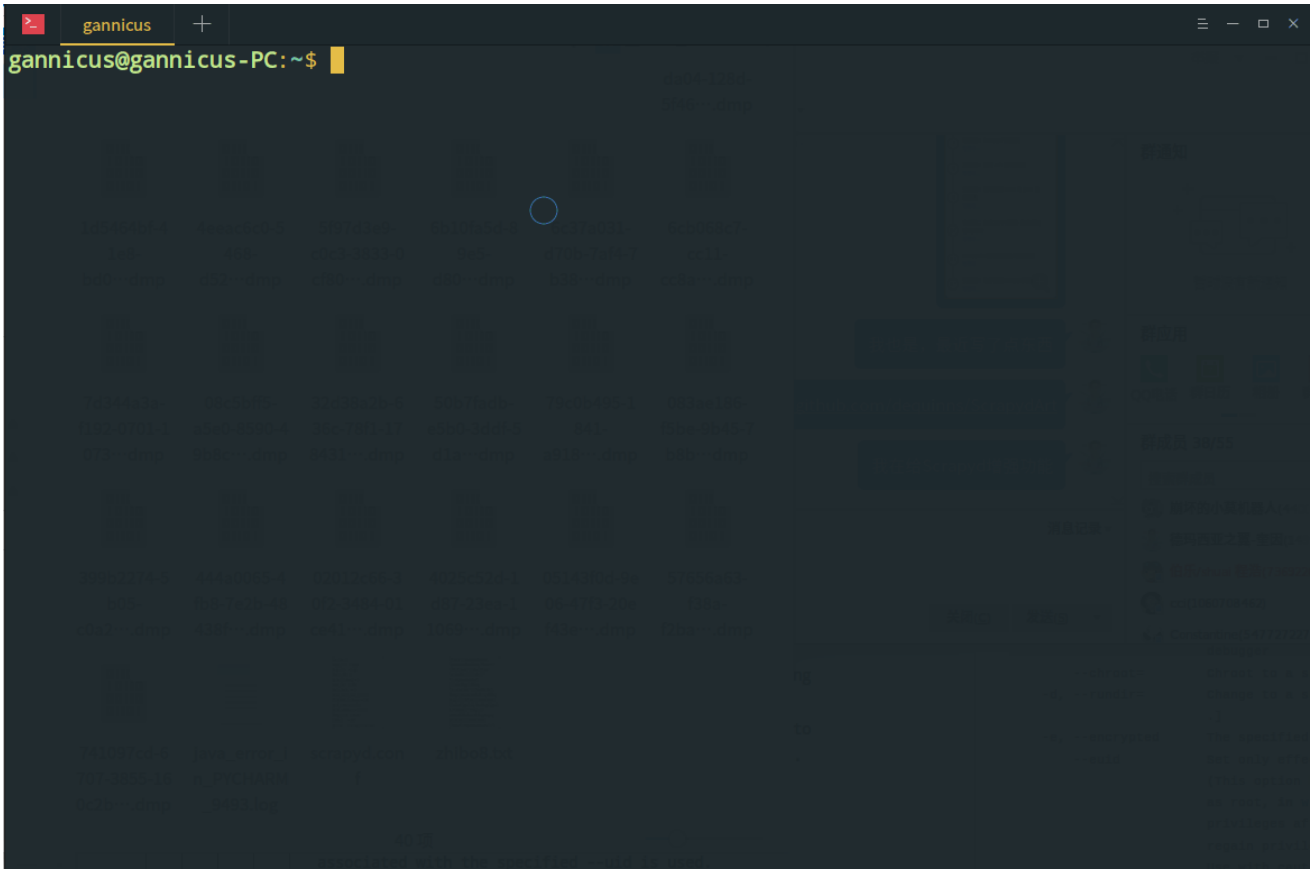
更新频率



从代码提交的统计图中可以知道，Scrapyd 的更新频率与 Scrapy 一样，都不高，并且改动也不大，这也为使用者节省了学习成本（前端开发环境一片混沌，新技术层出不穷）。

查看帮助

帮助信息中往往有一些令人惊喜的知识，而且这些知识很少在文档中出现。在命令行运行`scrapyd -h`可以获取 Scrapyd 的帮助信息：




```
Usage: twistd [options]
Options:
  -b, --debug          Run the application in the Python Debugger (implies
                       nodaemon),         sending SIGUSR2 will drop into
                       debugger
      --chroot=        Chroot to a supplied directory before running
  -d, --rundir=        Change to a supplied directory before running [default:
                       .]
  -e, --encrypted      The specified tap/aos file is encrypted.
      --euid           Set only effective user-id rather than real user-id.
                       (This option has no effect unless the server is running
                       as root, in which case it means not to shed all
                       privileges after binding ports, retaining the option to
                       regain privileges in cases such as spawning processes.
                       Use with caution.)
  -f, --file=          read the given .tap file [default: twistd.tap]
  -g, --gid=           The gid to run as.  If not specified, the default gid
                       associated with the specified --uid is used.
      --help           Display this help and exit.
      --help-reactors  Display a list of possibly available reactor names.
  -l, --logfile=       log to a specified file, - for stdout
      --logger=        A fully-qualified name to a log observer factory to use
                       for the initial log observer.  Takes precedence over
                       --logfile and --syslog (when available).
  -n, --nodaemon       don't daemonize, don't use default umask of 0077
  -o, --no_save        do not save state on shutdown
      --originalname   Don't try to change the process name
  -p, --profile=       Run in profile mode, dumping results to specified file.
```

```
      --pidfile=       Name of the pidfile [default: twistd.pid]
      --prefix=        use the given prefix when syslogging [default: twisted]
      --profiler=      Name of the profiler to use (profile, cprofile).
                       [default: cprofile]
  -r, --reactor=       Which reactor to use (see --help-reactors for a list of
                       possibilities)
  -s, --source=        Read an application from a .tas file (AOT format).
      --savestats      save the Stats object rather than the text output of the
                       profiler.
      --spew           Print an insanely verbose log of everything that happens.
                       Useful when debugging freezes or locks in complex code.
      --syslog         Log to syslog, not to file
  -u, --uid=           The uid to run as.
      --umask=         The (octal) file creation mask to apply.
      --version        Print version information and exit.
  -y, --python=        read an application from within a Python file (implies
                       -o)

twistd reads a twisted.application.service.Application out of a file and runs
it.
Commands:
    conch            A Conch SSH service.
    dns              A domain name server.
    ftp              An FTP server.
    inetd            An inetd(8) replacement.
    manhole          An interactive remote debugger service accessible via
                     telnet and ssh and providing syntax coloring and basic line
                     editing functionality.
    portforward      A simple port-forwarder.
    procmon          A process watchdog / supervisor
    socks            A SOCKSv4 proxy service.
    web              A general-purpose web server which can serve from a
                     filesystem or application resource.
    words            A modern words server
    xmpp-router      An XMPP Router server


```

以指定的 Scrapyd 配置文件启动

Scrapyd 的配置文件读取优先级在文档中有介绍：

```
Scrapyd searches for configuration files in the following locations, and parses them in order
with the latest one taking more priority:

    /etc/scrapyd/scrapyd.conf (Unix)
    c:\scrapyd\scrapyd.conf (Windows)
    /etc/scrapyd/conf.d/* (in alphabetical order, Unix)
    scrapyd.conf
    ~/.scrapyd.conf (users home directory)

```

大概意思为：Scrapyd 优先在以下位置搜索名为`scrapyd.conf`的配置文件，并按顺序解析它们，最新的配置文件具有更高的优先级。

根据上方帮助文档中`-d, --rundir= Change to a supplied directory before running [default:.]`的介绍，Scrapyd 还可以指定配置文件所在目录来启动。比如在`/home/gannicus/`目录下有一个名为`scrapyd.conf`的文件，在配置中笔者将端口改成`6900`，其他配置不变。然后笔者可以通过`-d`命令指定目录，也可以在`/home/gannicus/`目录中启动命令面板，它会默认在当前目录寻找`scrapyd.conf`文件。

> 下方的 GIF 动图展示了当前目录与指定目录启动 Scrapyd 的全程操作