[Tech Share]Log analyses tools and Usage

INTRODUCTION

This is apache log analysis tool, which can be got statics data such as average elapsed time of each type of url, counts , page elapsed time etc. That data can be trace page performance changes on product environment by apache log.

Overview of the project

This project cotains two tools: preprocess.py and tinylogan.py.

- preprocess.py mainly classify each url, such as url /plan/123 and /plan/124 are the same type url, only parameter are different. So it will replace them as /plan/${planid}. This program search the log line by line and replace the string according to the config file(default is matchPattern.cfg, can be change to other file), and save the changed log beside the orignal one with prefix 'res_' added
- tinylogan.py analyses the changed log provided by preprocess.py, and get out put of the summary of the log,inlcuding average time, calling times, total time of the distinguished urls and pages.
- runme automatically download the log file from specific url and analyse it ,then store the results to database

What's more, you can add -S option when run the tinylogan.py to save the results in database. There is also a supporting website developed in django whitch can show the results graphically and compare more than one result. Of course you should set up the website in advance follow the instrument provide later

Pre-requirement

To use this tool,change the configuration of log format from something(default config) like this:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

To this:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T/%D"
combined
```

The log record will change to something like this:

```
[31/Jan/2008:14:19:07 +0000] "GET / HTTP/1.1" 200 7918 ""
... "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.11) Gecko/20061201
Firefox/2.0.0.11 (Ubuntu-feisty)" 0/95491
```

Doploy

The follwing python lib should be installed in advance. It is suggested that a virtual envirenment should be created

- Django==1.6.5
- gunicorn==19.0.0
- lxml==3.3.5
- psycopg2==2.5.3
- pygal==1.4.6
- wsgiref==0.1.2

Also database PostgreSQL should be installed and configured properly in advance. This includes: create user with username '**redhat**' and password '**redhat**', and create database named '**log**'. Then run

the shell script run.py under deploy directory. If there is no error, the script wil create tables which store log analysis results.

Git clone the code from github.

The server can be start easily with the following command executed in the same directory with manage.py (IP should replace with your own):

```
gunicorn logView.wsgi:application --bind 10.66.136.238:8000 &
```

HOW TO USE

The easiest way to know how to use it :

```
command --help
```

Here goes some comman examples for each tool:

preProcess.py:

i.e. preprocess the data/0709.log with the config file 'tcms.cfg', and the result file will be stored in the same directory with a prefix 'res_'

```
python preprocess.py -c tcms.cfg data/0709.log
```

tinylogan.py

i.e. analyse the log located in data/res_ssl_access_log from 2014/07/01 to 2014/0702/ and save the results in database(-S)

```
python tinylogan.py data/res_ssl_access_log --start-date=1/Jul/2014
--end-date=1/Jul/2014 -S
```

runme.py

before add runme.py to your daily cron job, you should run this scripts with absolute path and check whether there are any errors.

Then you can add it to Cron jobs.

```
python /absolute/path/to/runme.py
```

RESULTS

Let explain the given results:

```
Starting from 15/Apr/2011:08:19:06
enough... stopped by user action
Ending at 28/Apr/2011:17:00:36
Elapsed time: 0:00:04.955008
Timedelta is 13 days, 8:41:30 (but only 7 days, 9:41:30 are counted due to time
bounds)
```

Starting from ...

First valid entry found in the log

enough... stopped by user action

Only if you CTRL+C during the log analysis. This will stop the log scan and skip to results immediatly

Ending at ...

Last entry analyzed

Elapsed time: ...

Time required for the log analysis

Timedelta is ...

Number of days from the first and last entry of the log, important for giving to the users a percent of the total time taken from an entry.

If you use some of the time filters above the used value for the statistic is the one given in the sentence but only xxx are counted due to time bounds.

The following showed in file ./log/res.txt:

```
Top page called times
  0001 - page1 3035.963 (1760 times, average 0.828, 18.47% of the total)
  0002 - page2 3035.963 (1760 times, average 0.828, 18.47% of the total)
  ...

Top page total time
  0001 - page3 139716.277 (1190 times, average 117.409, 32.69% of the total)
  0002 - page4 38809.064 (94 times, average 66.002, 9.08% of the total)
  ...

Top page average time
  0001 - page5 1791.993 (3 times, average 298.666)
  0002 - page6 259.292 (0 times, average 259.292)
  ...

Top total time
  0001 - /url1 46591.603 (4924 times, average 9.462, 7.28% of the total)
  0002 - /url2 12660.053 (1212 times, average 10.446, 1.98% GET/case/idnumberof the
total)
  ...

Top average time
  0001 - /url3 32.828 (15 times, 492 total)
  0002 - /url4 30.549 (7 times, 213 total)
  ...

Top call times
  0001 - /url5 1.014 (349028 times, 354008 total)
  0002 - /url6 1.122 (6564 times, 7364 total)
  ...
```

TODO

- Analyse the log with a URL online
- provide an easier way to do with the page configuration

FAQ

1. how to create 'redhat' user with password 'redhat' in PostgreSQL and create database 'log'?
    a. `open /var/lib/pgsql/data/pg_hba.conf, find 'peer' and change to 'md5' because of the different types of authentication`
    b. then connect to PostgreSQL with admin user, the default is psql
    c. execute following command:
        i. `CREATE USER redhat WITH PASSWORD 'redhat';`
        ii. `CREATE DATABASE log;`
        iii. `GRANT ALL PRIVILEGES ON DATABASE log to redhat;`
        iv. `FLUSH PRIVILEGES;`
2. how to install python lib `psycopg2`?
    a. `sudo yum install postgresql-libs`
    b. `sudo yum install postgresql-devel`
    c. `sudo yum install python-devel`
    d. `sudo pip install psycopg2`

NOTES

Thanks to @keul (github). tinylogan is his project and i modified and add some new features. This is the github link.