# Contents

1 Description  1
2 Architecture  3
3 Inside info  5
4 Install  7
   4.1 staLUN installation instructions  7
Getting statistics via EMC was quite painful/costly unless someone had purchased the corresponding tool/lib. But not anymore. staLUN is an open-source web platform that parses EMC generated csv files, stores the data in rrd files and graphs statistics about Read/Write Requests, Blocks Reads/Writes. As it is heavily based on the django-taggit library, the platform admin is given the option to tag the LUNs via Django admin interface. This results in LUN categorization in the frontend.

As there is some rrd analyzing involved, staLUN requires the numpy lib to be present. Once, done, You can find the installation details at the Install staLUN section.
staLUN assumes that you have access to your EMC devices. An abstraction of the architecture is:

EMC saves the desired stats in csv files at the server that staLUN runs. staLUN parses these files every five (5) minutes and fills the rrd files with data. Django, aided by the rrdtool lib generates the graphs. That simple!
Well, that was simple but staLUN does much more under the hood. This involves caching, a simple statistical analysis on the graphs, tagging, plus rendering all the above at a Bootstrap3 themed UI.
4.1 staLUN installation instructions

Below is an example installation process on a basic system with the following conditions:

- Standard Debian system
- Nginx as reverse proxy
- Gunicorn as WSGI server
- Mysql as database
- Memcached as caching system
- ProFTPd to receive data sent from the EMC

but all of the steps should be easily adapted for other setups. Note that python-django-taggit is only packaged in Debian Jessie and later - for Wheezy or earlier you must install it manually by e.g. easy_install

==== ON THE STALUN HOST ====

aptitude install python-django python-django-taggit python-rrdtool python-numpy python-memcache \ python-gevent python-simplejson python-django-south python-mysqldb \ nginx-light gunicorn mysql-server mysql-client memcached proftpd-basic git

[..set root mysql passwd when asked..]
[..set to run proftpd from inetd when asked..]

cd [..PARENT OF INSTALL DIR..]

git clone https://code.grnet.gr/git/stalun

cd stalun

cp settings.py.dist settings.py
$EDITOR settings.py
[..standard django edits, and ensure the password is the same as the one used in the database setup below, and $EMC_STATS_DIR is the same as in the ftpd setup below..]

cp apache/django.wsgi.dist apache/django.wsgi
$EDITOR apache/django.wsgi

mysql -u root -p << EOF
create database stalun;
create user 'stalun'@'localhost' identified by '[..PASSWORD.];';
grant all on 'stalun'.* to 'stalun'@'localhost' identified by '[..PASSWORD..]';
flush privileges;
EOF

./manage.py syncdb

./manage.py migrate

$EDITOR /etc/nginx/sites-available/stalun

==8<=

server {
    listen 80; ## listen for ipv4
    listen [::]:80; ## listen for ipv6
    server_name stalun.domain.com;
    root /usr/share/nginx/www;
    index index.html index.htm;
    access_log /var/log/nginx/stalun.access.log;
    error_log /var/log/nginx/stalun.error.log;
    location / {
        return 301 https://stalun.domain.com;
    }
    location /nginx_status {
        stub_status on;
        access_log off;
        allow 127.0.0.1;
        allow ::1;
        deny all;
    }
}

server {
    listen 443; ## listen for ipv4
    listen [::]:443; ## listen for ipv6
    server_name stalun.domain.com;
    access_log /var/log/nginx/stalun-ssl.access.log;
    error_log /var/log/nginx/stalun-ssl.error.log;
    ssl on;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;
    ssl_dhparam /etc/ssl/dhparams-2048.pem;
    ssl_certificate /etc/ssl/certs/stalun_domain_com-chained.crt;
    ssl_certificate_key /etc/ssl/private/stalun_domain_com.key;
    keepalive_timeout 60;
    # HSTS support on nginx, max-age 1 month
    add_header Strict-Transport-Security max-age=2592000;
    location /static {
        root /srv/stalun/public;
    }
    location / {
        proxy_pass http://127.0.0.1:8090;
        allow [...] EXAMPLE IPv4 RANGE...];
        allow [...] EXAMPLE IPv6 RANGE...];
        deny all;
    }
}

===>8==

Chapter 4. Install
ln -s /etc/nginx/sites-available/stalun /etc/nginx/sites-enabled/stalun

service nginx restart

$EDITOR /etc/gunicorn.d/stalun

==8<=

CONFIG = {
    'mode': 'django',
    'working_dir': '[..INSTALL_DIR..]',
    'user': 'www-data',
    'group': 'www-data',
    'args': {
        '--bind=127.0.0.1:8090',
        '--workers=2',
        '--timeout=30',
        '--debug',
        '--log-level=debug',
        '--log-file=/var/log/gunicorn/stalun.log',
    },
}

===>8==

service gunicorn restart

$EDITOR /etc/proftpd/conf.d/emc-stats.conf

==8<=

<Anonymous [..EMC_STATS_DIR..]>
  User ftp
  Group nogroup
  UserAlias anonymous ftp
  DirFakeUser on ftp
  DirFakeGroup on ftp
  RequireValidShell off
  MaxClients 10
  DisplayLogin welcome.msg
  DisplayChdir .message
  <Directory *>
      <Limit WRITE>
      DenyAll
      </Limit>
      </Directory>
  <Directory incoming>
      Umask 022 022
      <Limit READ WRITE>
      DenyAll
      </Limit>
      <Limit STOR>
      AllowAll
      </Limit>
      </Directory>
  </Anonymous>

===>8==

service proftpd restart

==== ON THE EMC ====

4.1. staLUN installation instructions
```bash
#!/bin/sh
HOMEDIR='/home/[..USER..]'
STATSDIR='emc-stats'
FTPHOSTS='[..STALUN_HOST..]'
FTPUPDIR='incoming/'
FULLDIR=$HOMEDIR/$STATSDIR
[ -d $FULLDIR ] || mkdir -p $FULLDIR
DATE='date +%s'
/nas/sbin/navicli -h spa getlun -name -brw -rwr >$FULLDIR/lun-stats.out-$DATE
echo $DATE >$FULLDIR/lun-stats.mid-$DATE
echo "LUN,Blocks Read,Blocks Write,Read Reqs,Write Reqs," >>$FULLDIR/lun-stats.mid-$DATE
sed -e '/LOGICAL UNIT NUMBER/d' 
   -e '/Name *LUN.*/d' 
   -e '/Name *Celerra_NS-480_*/d' 
   -e '/Blocks */d' 
   -e '/(Read|Write) Requests:/d' 
   $FULLDIR/lun-stats.out-$DATE >>$FULLDIR/lun-stats.mid-$DATE
cat $FULLDIR/lun-stats.mid-$DATE | sed -e '1n;2n;N;N;N;N;N;s/\n/,/g' | 
   egrep -v "0,0,0" >$FULLDIR/lun-stats.final.out-$DATE
rm -f $FULLDIR/lun-stats.out-$DATE
rm -f $FULLDIR/lun-stats.mid-$DATE
cd $HOMEDIR
for ftphost in $FTPHOSTS; do
  lftp $ftphost <<EOF
  cd $FTPUPDIR
  mput -c $STATSDIR/lun-stats.final.out-*
EOF
done
mv $FULLDIR/lun-stats.final.out-* $FULLDIR/sent/

#!/bin/sh
HOMEDIR='/home/[..USER..]'
STATSDIR='emc-stats'
FULLDIR=$HOMEDIR/$STATSDIR/sent
DATE='date +%Y%m%d'
find $HOMEDIR/$STATSDIR/sent -print0 -mtime +30 | 
  tar -czvf /$HOMEDIR/$STATSDIR/sent-monthly-$DATE.tar.gz -T -
find $HOMEDIR/$STATSDIR/sent -mtime +30 -delete

*/5 * * * * /home/[..USER..]/lun-stats.sh
30 3 1 * * /home/[..USER..]/lun-stats-archive.sh
```

Chapter 4. Install