Soothsayer Installation Guide

Overview

Soothsayer is split up into 4 main sections:

- Database
- Retriever
- Web Backend (Apache and CGI)
- Drupal
- Web Frontend (Simulations and Graphs)

Installation

The git repository for everything we created is located at:

```
http://projects.cs.uaf.edu/cs472_2014_spring_gi_aurora
```

Before we set up anything, we need to install all the software before hand, below is the list of things that need to be installed:

```
        drupal7
        mysql or mariadb
        python-mysqldb
```

Database Setup

Any *MySQL* compatible database can be used, we used *MariaDB 5.5*, which is a drop in replacement for *MySQL* (the binary is even called *mysql*).

The file below, located in the git repository, contains the template for the forecast database:

```
cs472_2014_spring_gi_aurora/db_template/forecast_db.sql
```

Create and initialize the database (which we called *forecast_db*) by doing the following from a terminal:

```
cd cs472_2014_spring_gi_aurora/db_template
mysql -u root -p -e "create database forecast_db"
mysql -u root -p forecast_db < forecast_db.sql
```

It is recommended that the forecast database username and password be different from the root username and password. This is just a precaution, but strongly recommended. The following will create a user named *forecast_user* with a password *letmein* to the database *forecast_db*:

```
mysql -u root -p
grant usage on *.* to forecast_user@localhost identified by 'letmein';
grant all privileges on forecast_db.* toforecast_user@localhost; flush privileges;"
```

Retriever Setup

Once the database is created, we can set up the scripts that retrieve data from NOAA and ALPO and insert that data into the database.

All the retriever scripts are located in the following folder in the git repository:

```
cs472_2014_spring_gi_aurora/src/forecast
```

All the scripts are written in Python 2.x. Install the following Python module:

```
python-mysqldb
```

The module above allows the retriever scripts to interface with the MySQL database.

Make sure all of the Python scripts have their executable flag set by doing the following in a terminal:

```
cd cs472_2014_spring_gi_aurora/src/forecast
chmod +x *.py
```

Before running any script, there is a configuration file that needs to be setup first, it is located in the git repository:

```
cs472_2014_spring_gi_aurora/src/forecast/forecast_retriever.cfg
```

Edit the configuration file with a text editor, it should contain the following:

```
[Contact Info]
receiver_email = Administrator Bob <bob@admin.net>

[Data Resources]
now_forecast_link = http://www.swpc.noaa.gov/ftpdir/lists/wingkp/wingkp_list.txt
h1_forecast_link = http://www.swpc.noaa.gov/ftpdir/lists/geomag/AK.txt
d3_forecast_link = http://www.swpc.noaa.gov/ftpdir/forecasts/geomag_forecast/%m%dgeomag_forecast.txt
d28_forecast_link = http://www.swpc.noaa.gov/ftpdir/weekly/27DO.txt
cr_link = http://alpo-astronomy.org/solar/rotn_nos.html

[Database]
forecast_database_name = forecast_db
forecast_database_user = user
forecast_database_password = password
```

Fill out the *receiver_email* field and the bottom three *forecast_database_** fields. The *receiver_email* field is the account to which all error messages are sent. The *forecast_database_** fields are the database and credentials used when data is downloaded and stored.

Note the *Data Resources* section in the configuration file. These are the sources on the net. Pay close attention to the *d3_forecast_link* field. Note the *%m%d* near the end of the field. This automatically fills in the current month (*%m*) and day (*%d*).

Once the fields above are filled out, the scripts should be tested to make sure the configuration file was set up correctly. Enter the following in a terminal:

```
sudo ./forecast_retriever.py -nhdmc
```

The above command downloads the now forecast (*n*), 1-hour forecast (*h*), 3-day forecast (*d*), 28-day forecast (*m*), and carrington rotation (*c*) data from the remote sources listed in the configuration file. Hopefully, the following text should appear:

```
Forecast Retriever
    Loading Configuration File          :)
    Retrieving Now Cast                 :)
    Retrieving Now Cast                 :)
    Retrieving 3 Day Cast               :)
    Retrieving 28 Day Cast              :)
    Retrieving Carrington Rotation:)
```

Next, we need to make sure the configuration file is secure. The best way to do this is to limit who can access the file. Do this by issuing the following commands in a terminal:

```
sudo chown root forecast_retriever.cfg
sudo chgrp www-data forecast_retriever.cfg
sudo chmod 640 forecast_retriever.cfg
sudo chattr +i forecast_retriever.cfg
```

The commands above make *root* the owner of the file. The group is set to *www-data*, this way the CGI scripts in next step can access the data stored in the database. Note, the *www-data* user cannot login by default, and they can only view files in a specified location. The *chmod* command allows *root* read and write permission to the file, read only permission to the *www-data*, and no permissions to anyone else. Finally, the *chattr +i* command makes the file immutable, so the file cannot be touched in any way, even by root, until the flag is removed.

Now that the retriever scripts are set up, we can now create a *cron* for each data retrieval. We decided to create scripts in a standard folder so they were easier to use, we did this in:

```
/home/crons
```

We then created 5 files:

nowcast.sh:

```
#!/bin/sh
cd cs472_2014_spring_gi_aurora/src/forecast
./forecast_retriever.py -n
```

h1cast.sh:

```
#!/bin/sh
cd cs472_2014_spring_gi_aurora/src/forecast
./forecast_retriever.py -h
```

d3cast.sh:

```
#!/bin/sh
```

```
       cd cs472_2014_spring_gi_aurora/src/forecast
       ./forecast_retriever.py -d
```

d28cast.sh:

```
#!/bin/sh
cd cs472_2014_spring_gi_aurora/src/forecast
./forecast_retriever.py -m
```

cr.sh:

```
#!/bin/sh
cd cs472_2014_spring_gi_aurora/src/forecast
./forecast_retriever.py -c
```

We decided to update the now forecast every 15 minutes, the 1 hour forecast every hour, the 3 day forecast every 3 hours, the 28 day forecast every day, and the carrington rotation every year. Do this by editing the *crontab* file for the *root* user:

```
sudo crontab -e
```

Add the following to the *crontab* file:

```
*/15 * * * *  /home/crons/nowcast.sh
@hourly             /home/crons/h1cast.sh
0 */03 * * *  /home/crons/d3cast.sh
@daily              /home/crons/d28cast.sh
@yearly             /home/crons/cr.sh
```

Web Backend Setup

First, start by enabling *Drupal*.

```
sudo cp /etc/drupal/7/apache2.conf /etc/apache2/mods-enabled/drupal.conf
```

Disable the default configuration file:

```
sudo rm -f /etc/apache2/sites-enabled/000-default.conf
```

Create a configuration file for the *Drupal* site by creating a file with your favorite text editor in:

```
/etc/apache2/sites-available/drupal7.conf
```

Copy and paste the following into the file created above:

```
<VirtualHost *:80>
       ServerAdmin bob@admin.net
       DocumentRoot /usr/share/drupal7/
       ServerName www.yoursite.com
       ServerAlias www.yoursite.com
       RewriteEngine On
       RewriteOptions inherit
</VirtualHost>
```

Enable the *Drupal* site configuration created above:

```
cd /etc/apache2/sites-available/
sudo ln -s drupal7.conf /etc/apache2/sites-enabled/drupal7.conf
```

To prevent an error from showing up in logs and such, issue the following command:

```
sudo echo "ServerName localhost" |sudo tee -a /etc/apache2/apache2.conf
```

Next, we will enable the CGI module:

```
sudo ln -s /etc/apache2/mods-available/cgi.load /etc/apache2/mods-enabled/
```

Restart *Apache* to apply the changes made so far:

```
sudo service apache2 restart
```

The CGI scripts can only be executed from a specific folder:
/usr/lib/cgi-bin/

We need to link the CGI files and the configuration file from the retriever folder directory:

```
sudo ln -s cs472_2014_spring_gi_aurora/src/forecast/cr_cgi.py /usr/lib/cgi-bin/
sudo ln -s cs472_2014_spring_gi_aurora/src/forecast/kp_cgi.py /usr/lib/cgi-bin/
sudo ln -s cs472_2014_spring_gi_aurora/src/forecast/forecast_retriever.cfg /usr/lib/cgi-bin/
```

Drupal Setup

Setup *Drupal* by opening a web browser and following the steps at the following address (note, server_address is the server's address you are on):

```
http://server_address/install.php
```

Now we will install all the modules that Soothsayer will require, run the following in a terminal:

```
sudo mkdir /etc/drupal/7/sites/all
sudo -rf /etc/drupal/7/sites/all/modules
sudo mkdir /etc/drupal/7/sites/all/modules
wget http://ftp.drupal.org/files/projects/ctools-7.x-1.4.zip
wget http://ftp.drupal.org/files/projects/date-7.x-2.7.zip
wget http://ftp.drupal.org/files/projects/entity-7.x-1.5.zip
wget http://ftp.drupal.org/files/projects/node_export-7.x-3.0.zip
wget http://ftp.drupal.org/files/projects/panels-7.x-3.4.zip
wget http://ftp.drupal.org/files/projects/rules-7.x-2.6.zip
wget http://ftp.drupal.org/files/projects/scheduler-7.x-1.2.zip
wget http://ftp.drupal.org/files/projects/services-7.x-3.7.zip
wget http://ftp.drupal.org/files/projects/simplenews-7.x-1.1.zip
wget http://ftp.drupal.org/files/projects/uuid-7.x-1.0-alpha5.zip
wget http://ftp.drupal.org/files/projects/views-7.x-3.7.zip
wget http://ftp.drupal.org/files/projects/views_infinite_scroll-7.x-1.1.zip
sudo unzip -o ctools-7.x-1.4.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o date-7.x-2.7.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o entity-7.x-1.5.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o node_export-7.x-3.0.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o panels-7.x-3.4.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o rules-7.x-2.6.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o scheduler-7.x-1.2.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o services-7.x-3.7.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o simplenews-7.x-1.1.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o uuid-7.x-1.0-alpha5.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o views-7.x-3.7.zip -d /etc/drupal/7/sites/all/modules
sudo unzip -o views_infinite_scroll-7.x-1.1.zip -d /etc/drupal/7/sites/all/modules
```

Now install the mobile theme we use:

```
sudo rm -rf /etc/drupal/7/sites/all/themes
sudo mkdir /etc/drupal/7/sites/all/themes
wget http://ftp.drupal.org/files/projects/rubix-responsive-theme-7.x-1.2.zip
sudo unzip -o rubix-responsive-theme-7.x-1.2.zip -d /etc/drupal/7/sites/all/themes
```

Since the Drupal pages and views are tedious to set up, use the site image we made:

```
cd cs472_2014_spring_gi_aurora/src/setup
sudo unzip -o site.zip -d /etc/drupal/7/sites/default/files/
mysql -u root -p drupal7 < drupal7.sql
```

The default administrator account and password is:

Username:	`admin`
Password:	`admin`

Web Frontend Setup

All the simulation files are located in the following directory in the git repository:

```
cs472_2014_spring_gi_aurora/src/simulation
```

The web frontend interface to Drupal is all set up from the image before, but the actual files we created must be located inside the web folder for the Drupal site. Copy or soft link the simulation folder above into the following folder:

```
/etc/drupal/7/sites
```

Examples:

```
sudo ln -s cs472_2014_spring_gi_aurora/src/simulation /etc/drupal/7/sites
```

or

```
sudo cp -r cs472_2014_spring_gi_aurora/src/simulation /etc/drupal/7/sites
```

Other Notes

If there are any questions not addressed in the document, contact:

Mike Moss <mrmoss@alaska.edu>