Redesign of the CERSER Cataloguing System for TeraScan® Processed Images

2010-2011 Multimedia Team

Autumn Luke, Patrina Bly
Center of Excellence in Remote Sensing Education and Research
Elizabeth City State University
Elizabeth City, North Carolina, USA

***Abstract* — The Center of Excellence in Remote Sensing Education and Research (CERSER) on the campus of Elizabeth City State University is currently tasked with the responsibility of receiving remotely sensed Advanced Very High Resolution Radiometer (AVHRR) data from orbiting National Oceanic and Atmospheric Administration (NOAA) satellites. This data is collected by the SeaSpace TeraScan® system installed in the CERSER labs in Dixon-Patterson Hall.**

When this system was initially installed in 2005, the data was collected, processed, annotated, and transformed into images in the Tagged Image File Format (TIFF) on a Windows® based server. These TIFF images were then uploaded to the CERSER archive library server located at http://cerser.ecsu.edu. Once uploaded, they were converted into various resolutions and their information was added to a tracking database maintained with Microsoft Access software. This database provided a searchable means for retrieving satellite image data through various parameters.

Since that time the CERSER server has been replaced with a Macintosh based server which cannot interact with the Microsoft based database and the Javascript scripting language. The goal of this project was to redesign the database and the code required to process the images. PHP, My Structured Query Language, and Command Line instructions to the software package ImageMagick® were utilized to complete these tasks.

Keywords-component; Terascan, ImageMagick, Macintosh OSX, MySQL, PHP, Remote Sensing

I. INTRODUCTION

The Center of Excellence in Remote Sensing Education and Research (CERSER) website was transferred from a Windows operating system to a Macintosh Operating system during the fall of 2010. This website stores and processes images produced from data received from NOAA satellites. The TeraScan system used to create these Tagged Image File Format (TIFF) images places them onto the CERSER server where they will be converted to Joint Photographic Experts Group (JPEG) format and resized. The script also adds the files to a database for later retrieval. This script was originally written in 2006 to utilize Javascript scripting along with Microsoft's Access database software. Due to the change in operating systems, this combination is no longer functional.

The goal of this project was to develop a new scripting file utilizing PHP, ImageMagick®, and the My Structure Query Language (MySQL) to process these images and add records to the database. These software packages are cross-platform compatible and Open Source which makes any future changes in operating systems more straightforward to implement.

II. SOFTWARE AND LANGUAGES

A. *ImageMagick®*

One of the major segments of this project was to convert and resize processed TIFF images from the TeraScan system. To complete this process, the software suite ImageMagick® was employed. This software is used to edit, create, and convert images in a range of formats including TIFF, JPEG, GIF, PNG, and several others. ImageMagick can also be used to distort images through resizing, cropping, flipping, and additional transformations.

ImageMagick was chosen for this project due to its ability to be executed from the command line. The software was used in the initial development in 2006 on the Windows platform to interact with Active Server Pages (ASP) and the TeraScan images. The transition from the Windows platform to the MAC OSX and PHP combination was accomplished smoothly due to the cross platform capabilities of ImageMagick. Utilizing the "exec()" PHP command, compiled instructions were able to be executed with two lines of code; one to compile the instructions into a variable and the next to execute it. The ImageMagick executable module "convert.exe" was the only module needed for this project, but the capability for further image manipulations remains available.

Another motivation for using this software was its availability as a free, Open Source, package. The complete installation process involved downloading and uncompressing the package and the setting of three system path variables. Once this was completed, directory permissions were modified using the "chmod" command allowing ImageMagick to write files to these directories.

B. *PHP*

PHP Hypertext Processor (PHP) is a programming language used to creating dynamic websites and was used in this Multimedia Team project. PHP can access a command line interface and perform specific tasks such as connecting and disconnecting to a database, copying files, and joining strings together. PHP is cross platform compatible and an Open Source language.

This project also utilized PHP to access ImageMagick through the Command Line Interface. ImageMagick was used to convert the TeraScan TIFF images into a JPEG format. ImageMagick converts images through the module "convert.exe".

C. MySQL

My Structured Query Language (MySQL) is an open-source relational database system owned by MySQL AB, a subordinate company of Oracle. MySQL operates as a server supplying multi-user access to multiple databases through MySQL commands.

The PHP command "mysql_query( )" performs a query and places the result into the variable "$result" for troubleshooting purposes. This project also used the MySQL "INSERT INTO" command to insert values into fields in a table of a database. There were two tables in this project that had values inserted in them. The two tables were the "images" table and the "CUpdate" table in the database TS2011. In the "images" table, the values were inserted into the fields: imagedate, time, satellite, product, description, and event. In the "CUpdate" table the variable was inserted into the field "updated".

D. Dreamweaver

To construct the PHP code and database connections needed for this project the web development software, Adobe Dreamweaver, was utilized. This package is available on multiple operating system platforms and offers the choice of a coding view or What-You-See-Is-What-You-Get (WYSIWYG) design view. This project relied solely on the coding view to create database connections and to utilize the PHP code elements. Database connections were created through the database window to the MySQL database on the CERSER server. Recordsets were formed through the Server behavior window.

The Dreamweaver coding view is color coded to differentiate between HTML, PHP, Comments, and other web development languages. Coding hints are given as the user types to assist in standard development. Once a command is started, Dreamweaver offers selections along with the options needed for that command.

E. Macintosh Scheduler

Scheduler for Macintosh is a scheduling application used to automate the launching of applications or the opening of documents at a set time. This action can occur on a precise date or time, at computer start-up, or after a period of idleness.

The scheduler was used for this project to run the auto_img_2011.php file each morning at 3:00 am. The scheduler wizard automates this process allowing the user to specify the time, file to open, and application to use.

III. The Code

A. Disassembling the 2006 Code

The first step in redesigning the conversion code was to disassemble the code developed in 2006. The code was broken down line by line and described using pseudocode, a method of describing code in plain English. The structure/process of the 2006 code was used as a base for the design process allowing the team to focus more on the PHP syntax to be used in building the scripts.

B. Connecting to the Database

Creating the image conversion file starts with connecting to the MySQL database. This is accomplished through a connection file created by Dreamweaver. The image conversion file calls this file in the first line establishing the tie to the database for further queries later in the code. The connection passes the user name, password, and the name of the database to the server to establish this connection.

The next section of code to be described creates the recordsets to be used in this file. The first recordset is composed of the records from the table "images" of the TeraScan MySQL database. All the records are chosen and ordered in ascending order by the "id" field. The second recordset is created from the "CUpdate" table and ordered in a descending order by the "id" field.

C. Checks Made Before Conversion

Several checks and variable assignments need to be made before the conversion loop is entered. The first step in the conversion process is to assign the directory containing the TeraScan processed TIFF images to the variable $dir. This variable will be used throughout the conversion process to access the original image to be handled and converted. This command is performed through the command: $dir = "original/".

To ensure that the contents of the variable $dir is a directory, the PHP command "is_dir()" is used to test it. This takes the form of: if (is_dir ($dir)) and returns true if the contents is a directory and false if it is not.

The next command, "opendir()", assigns the directory path to the variable $dh. This path will be used with the command "readdir()" to read the file names into a variable in the step. If the path is not valid, the command returns FALSE exiting out of the "if" statement. The syntax used for this step is: if ($dh = opendir ($dir)).

D. Beginning the Loop

The "readdir()" command returns the filename of the next file from the directory stored in $dh. The image filenames are returned in the order in which they are stored by the filesystem. This is the beginning of the looping process meant to process each TeraScan image in the original directory. The name of the file is placed into the variable $satName to be converted and added to the database. When all the files have been handled, the "readdir()" command returns FALSE which triggers the procedure to exit the loop. This line of code is constructed as: while (($satName = readdir ($dh)) !== false).

After obtaining the file name it is checked to ensure that it is a TIFF file that can be processed. The sub-string command "substr()" is used to retrieve the last four characters to ensure they are ".tiff". The script below illustrates this check.

```
if (substr ($satName, 15,5)=='.tiff' )
```

E. File Name Deconstruction

The first part of the image loop is to deconstruct the file name placing data into variables for later inclusion into the database. The command used in this process is substr (string $string , int $start [, int $length]) where the portion of the string returned is specified by the start and length parameters.

The first piece of information to be retrieved is the date and time that the image was created. This is found in characters 1-6 of the file name. The date is converted from the format YYMMDD to the format MM/DD/YYYY and placed into the $cnvrtDate variable. The time is placed into the variable $cnvrtTime. Identifying separate characters within the file name starts with the count of "0" and not "1". The structure of these commands is found below.

```
$cnvrtDate =substr ($satName, 2,2) . "/" . substr ($satName, 4,2) . "/20" . substr ($satName, 0,2);
$cnvrtTime =substr ($satName, 7,4);
```

The next step in the file name deconstruction is to obtain the satellite name from which the data was received to create the TIFF image. There are several NOAA satellites used at this time and the potential to receive SeaWiFS satellite data is still open if that contract was renewed. The same PHP command used to obtain the date, substr(), is used in conjunction with two "if" statements to determine which type of satellite was used and the number for that satellite if applicable. The name of the satellite is then placed into the variable $sat. If the satellite cannot be identified, "000" is placed into the variable. The "if" statements and the string commands are found below.

```
if (substr ($satName, 12, 1) == 'n' )
$sat = "NOAA-" . substr ($satName, 13, 2);
else
$sat= "000" ;
if (substr ($satName, 12, 1) == 's' )
$sat = "SeaWiFS" ;
```

F. Database Insertion

Now that the date, time, and satellite name have been placed into their respective variables, the information can be placed into the TeraScan database. The values are arranged in order as to the database field names in the table "images". The SQL statement begins with the command "INSERT INTO", creating a new record in the database. The entire SQL string is placed into the variable $query. The next line of the PHP code utilizes the command "mysql_query()" to execute the SQL statement in $query. The result of this command is placed into the variable $result in order to provide feedback to the user if the record insertion fails. The next PHP command, "mysql_insert_id()", retrieves the ID generated by the last query. This ID number is placed into the variable $lastID to be used later in naming the files. The three lines used for the above process are shown below.

```
$query = "INSERT INTO images (imgdate, time, satellite, product, description, event) VALUES ('" . $cnvrtDate . "', '". $cnvrtTime . "', '" . $sat . "',0 , '', 0)";

$result = mysql_query ($query );

$lastID = mysql_insert_id ();
```

G. Copy, Convert, Resize, Rename

Now that the database entry has been made, the original TIFF images can be converted, copied, and resized. The first step is to copy the original file using the standard PHP command "copy()". The location and name of the original file are placed into the variable $orgFile utilizing the script "$orgFile = $dir . $satName;". The new location and file name are placed into the variable $newfile utilizing the ID number in the variable $lastID as the new file name. The "copy" command is then issued from within an "if" statement which checks the completion of the "copy" command. This process is shown in the following lines.

```
$orgFile = $dir . $satName;

$newfile = 'actual_2011/' . $lastID . '.tiff' ;

if (!copy ($orgFile, $newfile )) {
echo "failed to copy $file...\n <br><br>" ;
}
```

The next procedure is to convert the original TIFF file to a JPEG file with no reduction in pixel file size. This will produce a smaller file in memory allowing for a faster download for users. The ImageMagick module convert.exe is used to accomplish this conversion. The location of this module, the contents of the variable $orgFile, and the location and name of the new file are placed into the variable $cnvrt. This line is then executed using the PHP command "exec()". This command operates in the same fashion as typing the command into the command line interface. The two lines below convert the TIFF file into a JPEG file and rename it using the number in the variable $lastID.

```
$cnvrt = '/opt/local/bin/convert ' . $orgFile . ' medium_2011/' . $lastID . '.jpg' ;

exec($cnvrt );
```

The next two lines below perform in the same way as the previous two lines, but also resize the TIFF image by 50% for a lower resolution image.

```
$cnvrt = '/opt/local/bin/convert ' . $orgFile . ' -resize 50% low_2011/' . $lastID . '.jpg' ;

exec($cnvrt );
```

A thumbnail must also be produced to be viewed on the web page. This is done in much the same method as the last two conversions, but utilizes other attributes to ensure the image is not distorted. These are the two lines used to accomplish this procedure.

```
$cnvrt = "/opt/local/bin/convert -size 120x120 '" . $orgFile . "' -thumbnail 120x120^ -gravity center -extent 120x120 'thumbs_2011/"
. $lastID . ".jpg'" ;

exec($cnvrt );
```

Once the conversion is complete, the script deletes the original file before proceeding to the next TIFF image. This is done utilizing the PHP command "unlink()". This is formatted as: unlink ($orgFile);, where the variable $orgFile holds the name of the original TIFF image.

The script now returns to the "while" statement to check for the next file to be processed. Once all of the files have been processed, the script closes the directory through the command: closedir ($dh);.

H. Update Time

Once the loops have completed, the script then inserts an update time into the CUpdate table of the database. This is performed by using the PHP command "date()" and the desired formatting of the date/time. The values, along with the SQL command, is inserted into the variable $query and executed using the PHP command "mysql_query()" with the result being checked for completion. The following three lines compile the data and execute it.

```
$timeNow = date('Y-m-d H:i:s' );

$query = "INSERT INTO CUpdate (updated) VALUES ('" .
$timeNow . "')" ;

$result = mysql_query ($query );
```

The memory of the two recordsets is now released using the PHP command "mysql_free_result()".

IV. Results

The conversion code created was tested with multiple sets of images using sample directories. It was quickly found that the script would run, but not complete the copy and convert operations. Once the permissions were changed for these directories, the code completed the operations without failing. When the testing was completed, the actual directory names needed were substituted into the code. The code was then executed with no failures.

V. Conclusion

The code developed during this project successfully fulfilled the goals set in the beginning. Images developed by the TeraScan system are now being converted on a daily basis by the developed file. Due to the fact that the file is cross platform compatible, future platform changes will be less difficult to implement.

VI. Future Work

A. Email Notification

Feedback from the daily execution of the file was coded to echo the update time and date to the screen. Future modifications could include a section which would send an Email notification to the database manager confirming the success of the update. This notification could also include the database values that were updated providing a complete representation of the processed files.

Future Multimedia Teams can use the Postfix Enabler software to send email updates. Postfix Enabler allows you to turn on Simple Mail Transfer Protocol Authentication (SMTP-AUTH) on the server, so that it can authorize remote users who need to send mail through it. The Postfix Enabler is not a free software; it can be purchased through the website www.macupdate.com/app/mac/13237/postfix-enabler for $9.99.

B. Image Presentation

Due to recent upgrades to the CERSER web site, the TeraScan images are no longer presented to users. A presentation of these images and the supporting data should be implemented on either the home page of CERSER in a limited manner (one or two images at a time) or on a separate page where more images can be displayed at one time. This can be accomplished through Dreamweaver utilizing the built in database functions.

References

[1] Scheduler for Macintosh v. 5, Retrieved March 22, 2011 from the World Wide Web, http://www.macscheduler.net/download.html.

[2] Sequel-Pro - Project Hosting on Google Code, Retrieved January 14, 2011 from the World Wide Web, http://code.google.com/p/sequel-pro/.

[3] PHP: MySQL – Manual, Retrieved January 14, 2011 from the World Wide Web, http://php.net/manual/en/book.mysql.php.

[4] Managing MySQL on Mac OS X - O'Reilly Media, Retrieved January 24, 2011 from the World Wide Web, http://oreilly.com/pub/a/mac/2005/12/13/mysql.html.

[5] Date and Time Formatting in MySQL and PHP, Retrieved March 5, 2011 from the World Wide Web, http://www.eltcalendar.com/stuff/datemysqlphp.html.

[6] PHP: ImageMagick – Manual, Retrieved February 12, 2011 from the World Wide Web, http://www.php.net/manual/en/book.imagick.php.

[7] Using Apache and PHP on Mac OS X, Retrieved February 16, 2011 from the World Wide Web, http://www.devarticles.com/c/a/Apache/Using-Apache-and-PHP-on-Mac-OS-X/.

[8] ImageMagick, Retrieved March 2, 2011 from the World Wide Web, http://www.imagemagick.org/script/index.php.

[9] 2005-2006 Multimedia Team, retrieved January 17, 2011 from the World Wide Web, http://nia.ecsu.edu/onr/05-06/researchteams/mmt/images/060406_0506_mmt_paper.pdf