LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
-LIGO-
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note **LIGO-T030291-00-C** 12/03/03

ROBO SCIMON

Chethan Parameswariah

This is an internal working note
of the LIGO Project.

LIGO Livingston Observatory
19100 Ligo Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352-0159
Phone (509) 372-8106
Fax (509) 372-8137

California Institute of Technology
LIGO Project – MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu

1. ABSTRACT

This document describes the working of the automatic lock statistics generation system called "ROBO SCIMON" and logging them into the LLO elog at 16:00 hours UTC every morning during science and engineering runs. Robo-scimon is part of the collection of software robots - "*SOFT-ROBOTS*" now working at LLO to ease and improve efficiency.

2. INTRODUCTION

Automation of LIGO CDS Systems is essential to maintain consistency and to minimize human errors, with the advancement of interferometer into low noise commissioning and science runs.

ROBO SCIMON is a software robot that generates interferometer daily lock statistics for previous 24 hours; weekly lock statistics for the previous week on Sunday's only and daily cumulative lock statistics from the beginning of science run till today and then save it as a text file with the correct name format. The information generated is then elogged to current day's elog at 16:00 hrs UTC during the science and engineering runs.

3. OPERATION

ROBO SCIMON is part of "*SOFT-ROBOTS*" - a collection of software programs and scripts called '*software robots'* that are run at LIGO to automate various duties of the scimons, operators and engineers to ease and improve efficiency while maintaining reliability and consistency on a daily basis.

The main program "autostat_elog3.pl" is a lock statistics generator and auto e-logger program. This is a Perl program that is spawned as a cronjob every morning at 10.00 AM (11:00 AM – during daylight saving time). This generates the interferometer daily lock statistics for the previous 24 hours from 16:00 UTC today to 16:00 UTC yesterday and the cumulative lock statistics from 16:00 UTC of the first day of the beginning of science run till 16:00 UTC today. The data obtained is saved as ASCII text with the correct file name format i.e., with yesterday's date and month appended to the filename. This is then e-logged on today's page of the e-log with the daily lock statistics shown in full as saved in the text file while the cumulative lock statistics is a link to the text file. On Sunday's the program also creates a weekly locked statistics file and also elogs it with the daily and cumulative lock statistics entry. Currently this cronjob is run on control3 in the mass storage room at LLO and the program requires this sun workstation to be 'ON' for it to work.

The program listing for autostat_elog3.pl (version 1.0) is in Appendix.

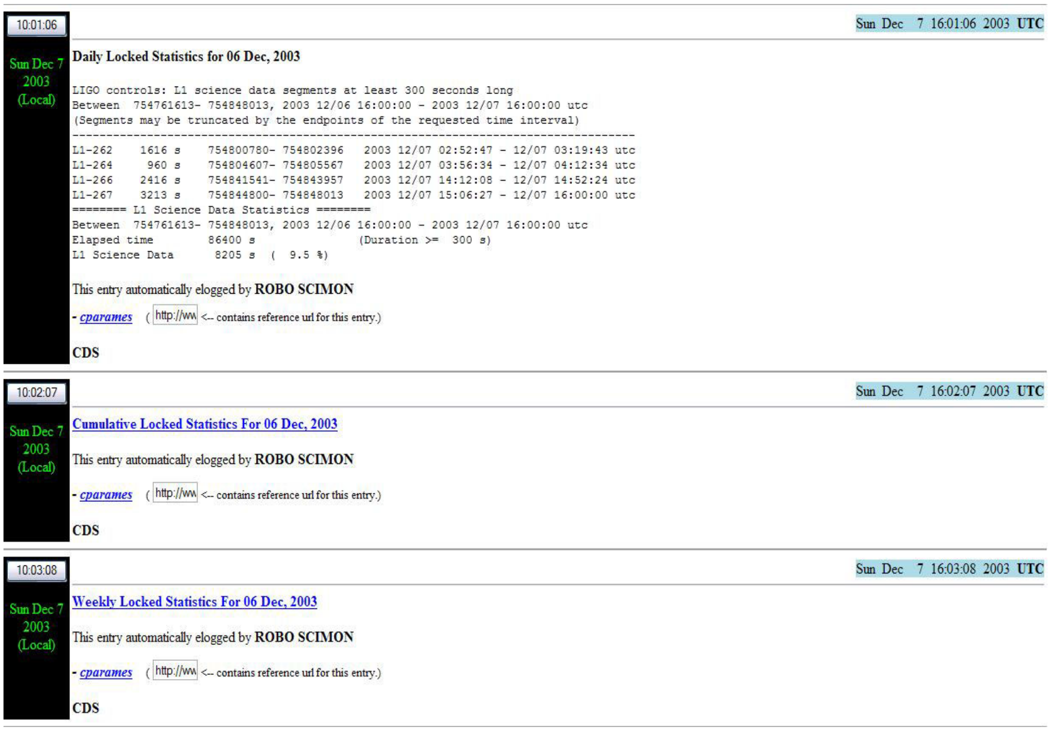
The perl program autostat_elog3.pl first has a set of variables hard coded into the program such as the location of required perl modules, science run start date, IFO identifier, and then calculates today's, yesterday's and last week's day, month and year information.

With this information, the conlog_science program (written by Peter Shawhan at Caltech) which produces the lock statistics as the output is spawned with the right arguments such as ifo identifier, minimum time for science segments, start and end times and the file name to be saved.

The files saved are then pre-pended and appended with html headers and footers respectively and saved with the day and month information included in the file name.

The perl program then elogs the full text of the daily lock statistics with links to both cumulative and weekly lock statistics on today's LLO elog.

Figure 1 shows the posting of the lock statistics on the LLO elog done during science run S3.



Daily Locked Statistics for 06 Dec, 2003

```
LIGO controls: L1 science data segments at least 300 seconds long
Between  754761613- 754848013, 2003 12/06 16:00:00 - 2003 12/07 16:00:00 utc
(Segments may be truncated by the endpoints of the requested time interval)
------------------------------------------------------------------------------------
L1-262    1616 s    754800780- 754802396   2003 12/07 02:52:47 - 12/07 03:19:43 utc
L1-264     960 s    754804607- 754805567   2003 12/07 03:56:34 - 12/07 04:12:34 utc
L1-266    2416 s    754841541- 754843957   2003 12/07 14:12:08 - 12/07 14:52:24 utc
L1-267    3213 s    754844800- 754848013   2003 12/07 15:06:27 - 12/07 16:00:00 utc
======== L1 Science Data Statistics ========
Between  754761613- 754848013, 2003 12/06 16:00:00 - 2003 12/07 16:00:00 utc
Elapsed time        86400 s               (Duration >=  300 s)
L1 Science Data      8205 s  (  9.5 %)
```

This entry automatically elogged by **ROBO SCIMON**

- *cparames* (http://ww <-- contains reference url for this entry.)

CDS

Cumulative Locked Statistics For 06 Dec, 2003

This entry automatically elogged by **ROBO SCIMON**

- *cparames* (http://ww <-- contains reference url for this entry.)

CDS

Weekly Locked Statistics For 06 Dec, 2003

This entry automatically elogged by **ROBO SCIMON**

- *cparames* (http://ww <-- contains reference url for this entry.)

CDS

Figure 1: Lock statistics entry automatically posted elogged by ROBO SCIMON.

Clicking on the link opens up the cumulative and weekly lock statistics in ascii text format on the browser. Figure 2 shows a daily summary plot for a one day during S3.

4. CONCLUSION

The ROBO SCIMON was initiated for the science run S3 and has been working since. Features such as operator start and stop, day light savings time identification, and the science run start and stop date information is currently either non-existant or is hard-coded in the program. The next version of the Robo-scimon will automate this adding intelligence to the program.

APPENDIX

Note: All the programs and configuration file are located on LLO CDS machines in the directory - */cvs/cds/project/roboscimon/*. The output text files are located in the directory - */opt/LLO/c/ops/public_html/S3/DailyStatistics/*.

1) Main perl program run as cronjob at 10:00 AM (11:00 AM during daylight saving time) – autostat_elog3.pl

```
#!/opt/apps/perl_5.6.1/bin/perl

#
# Rel 1.0 # First release : Chethan Parameswariah, LLO
# Rel 2.0 # Changed the script to actually elog using PERL HTTP
instead of netscape
# 27 Oct 2003
#

# Need these lib modules - this prepends to @INC at run time
#
#use lib "/opt/apps/perl_5.6.1/modules/HTML-Parser-
2.22/blib/lib";
#use lib "/opt/apps/perl_5.6.1/modules/libwww-perl-5.42/lib";
#use lib "/opt/apps/perl_5.6.1/modules/URI-1.02";
#use lib "/opt/apps/perl_5.6.1/modules/HTML-Parser-2.22/lib";
#use lib "/opt/apps/perl_5.6.1/modules/MIME-Base64-
2.11/blib/lib";

# Use this path and install all the modules in the project area
# Need these lib modules - this prepends to @INC at run time
#
use lib "/cvs/cds/project/roboscimon/modules/HTML-Parser-
2.22/blib/lib";
use lib "/cvs/cds/project/roboscimon/modules/libwww-perl-
5.42/lib";
use lib "/cvs/cds/project/roboscimon/modules/URI-1.02";
use lib "/cvs/cds/project/roboscimon/modules/HTML-Parser-
2.22/lib";
use lib "/cvs/cds/project/roboscimon/modules/MIME-Base64-
2.11/blib/lib";

# Tell what modules to use
#
use HTTP::Request::Common qw(POST);
use LWP::UserAgent;
use CGI;

# Set variables
#
$DEBUG = 1;
$dailytempfile = "/cvs/cds/llo/logs/dailystat.temp";
$weeklytempfile = "/cvs/cds/llo/logs/weeklystat.temp";
$cumtempfile = "/cvs/cds/llo/logs/cumstat.temp";
$CONLOG_PATH = "/cvs/cds/llo/conlog/bin/conlog_science";
$HTML_FILES_PATH =
"/opt/LLO/c/ops/public_html/S3/DailyStatistics";
$username = "cparames";
$password = "wave\$";

# Set the Science run start date at the begining of each run
$Srun_start = "2003/10/31,16:00:00";

# check for Day light savings time and if yes sleep for an hour
before running
($DST) = (localtime(time)) [8];
if ($DST) {
sleep 3600;
}

# Get yesterday's date, month and year
($DAY1_NUMBER, $MONTH1_NUMBER, $YEAR1_NUMBER) = (localtime(time-
86400)) [3,4,5];
$MONTH1_STRING = (qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov
Dec))[(localtime(time-86400)) [4]];
$MONTH1_NUMBER += 1;
if ($MONTH1_NUMBER < 10) {
   $MONTH1_NUMBER = "0".$MONTH1_NUMBER;
}
if ($DAY1_NUMBER < 10) {
   $DAY1_NUMBER = "0".$DAY1_NUMBER;
}
$YEAR1_NUMBER += 1900;

# Get lastweek's date, month and year
($DAY2_NUMBER, $MONTH2_NUMBER, $YEAR2_NUMBER) = (localtime(time-
604800)) [3,4,5];
$MONTH2_STRING = (qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov
Dec))[(localtime) [4]];
$DAY2_STRING = (Sun, Mon, Tue, Wed, Thu, Fri, Sat)[(localtime)
[6]];
$MONTH2_NUMBER += 1;
if ($MONTH2_NUMBER < 10) {
   $MONTH2_NUMBER = "0".$MONTH2_NUMBER;
}
if ($DAY2_NUMBER < 10) {
   $DAY2_NUMBER = "0".$DAY2_NUMBER;
}
$YEAR2_NUMBER += 1900;

# Get today's date, month and year
($DAY_NUMBER, $MONTH_NUMBER, $YEAR_NUMBER) = (localtime(time))
[3,4,5];
$MONTH_STRING = (qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov
Dec))[(localtime) [4]];
$MONTH_NUMBER += 1;
if ($MONTH_NUMBER < 10) {
   $MONTH_NUMBER = "0".$MONTH_NUMBER;
}
if ($DAY_NUMBER < 10) {
   $DAY_NUMBER = "0".$DAY_NUMBER;
}
$YEAR_NUMBER += 1900;

if ($DEBUG) {
print "Month = $MONTH2_STRING - $MONTH2_NUMBER\tDay =
$DAY2_NUMBER,\tYear = $YEAR2_NUMBER\n";
print "Month = $MONTH1_STRING - $MONTH1_NUMBER\tDay =
$DAY1_NUMBER,\tYear = $YEAR1_NUMBER\n";
print "Month = $MONTH_STRING - $MONTH_NUMBER\tDay =
$DAY_NUMBER,\tYear = $YEAR_NUMBER\n";
}
#use this if you want to test the script for a particular day ,
change the value.
#$DAY_NUMBER = 3;

# Delete the previous daily temp file
$command = "/bin/rm /cvs/cds/llo/logs/dailystat.temp";
system($command);

# Delete the previous cumulative temp file
$command = "/bin/rm /cvs/cds/llo/logs/cumstat.temp";
system($command);

if ($DAY2_STRING eq 'Sun') {
# Delete the previous weekly temp file
$command = "/bin/rm /cvs/cds/llo/logs/weeklystat.temp";
system($command);
}


# Set the variables to the dates
$Yesterday =
"$YEAR1_NUMBER/$MONTH1_NUMBER/$DAY1_NUMBER,16:00:00";
$Weekly = "$YEAR2_NUMBER/$MONTH2_NUMBER/$DAY2_NUMBER,16:00:00";
$Today = "$YEAR_NUMBER/$MONTH_NUMBER/$DAY_NUMBER,16:00:00";

#Parse the conlog for yesterdays and from start of run science
data and save it to a file
$command = "$CONLOG_PATH L1 300 $Yesterday $Today utc >
$dailytempfile";
```

```
if ($DEBUG) {
   print "$command \n";
}
system($command);
$command = "$CONLOG_PATH L1 300 $Srun_start $Today utc >
$cumtempfile";
if ($DEBUG) {
   print "$command \n";
}
system($command);

if ($DAY2_STRING eq 'Sun') {
$command = "$CONLOG_PATH L1 300 $Weekly $Today utc >
$weeklytempfile";
if ($DEBUG) {
   print "$command \n";
}
system($command);
}
$header = "/cvs/cds/project/roboscimon/conlogheader.txt";
$footer = "/cvs/cds/project/roboscimon/conlogfooter.txt";
$Yest_day = "$MONTH1_NUMBER$DAY1_NUMBER";
$Week_day = "$MONTH2_NUMBER$DAY2_NUMBER";

$command = "cat $header $dailytempfile $footer >
".$HTML_FILES_PATH."/LockStatistics_".$Yest_day."_l1.html";
if ($DEBUG) {
   print "$command \n";
}
system($command);

$command = "cat $header $cumtempfile $footer >
".$HTML_FILES_PATH."/LockStatistics_".$Yest_day."_l1_cum.html";
if ($DEBUG) {
   print "$command \n";
}
system($command);

if ($DAY2_STRING eq 'Sun') {
$command = "cat $header $weeklytempfile $footer >
".$HTML_FILES_PATH."/LockStatistics_".$Week_day."_week_l1.html";
if ($DEBUG) {
   print "$command \n";
}
system($command);
}

###### Auto Elog ######
# Create a new user agent
#
$ua = LWP::UserAgent->new();


# Since I need a proxy at LLO, set the proxy here
#
$ua->proxy('http','http://london.ligo-la.caltech.edu:80/');
```

```
# set url of elog
#
my $URL = 'http://www.ligo-la.caltech.edu/ilog/pub/ilog.cgi?';

# Elog file date
#
$log_file_date = "$MONTH_NUMBER/$DAY_NUMBER/$YEAR_NUMBER";
#$log_file_date = "12/24/2001";

# Put the file contents to a array variable list_text
open(DIAG,$dailytempfile)||die "Cannot open file
$dailytempfile\n";
@list_text = <DIAG>;
close DIAG;

$num = 1;
$comment_string = "";
foreach $line (@list_text)
{
 chomp $line;
 $comment[$num] = "$line"."<BR>";
 chomp $comment[$num];
 $comment_string = "$comment_string"."$comment[$num]";
 $num += 1;
}
 if ($DEBUG) {
 print "$comment_string \n";
 }

$Yesterday_notime = "$DAY1_NUMBER $MONTH1_STRING, $YEAR1_NUMBER";

$comment_string_header = "<b>Daily Locked Statistics for
$Yesterday_notime</b><BR><pre>";
$comment_string_footer = "</pre>This entry automatically elogged
by <b>ROBO SCIMON</b><BR>";
$comments =
$comment_string_header.$comment_string.$comment_string_footer;
if ($DEBUG) {
   print $comments."\n";
}

# POST it with contents and values
#
my $request = POST $URL,
        Content_Type => 'multipart/form-data',
        Content =>
        [
         group => 'detector',
         task => 'makeEntry',
         log_file_date => $log_file_date,
         comments => $comments,
         keywords => 'CDS',
         priority => 'normal',
         entry_author => 'cparames',
         'submit' => 'Submit Log Entry'
        ];
```

```
# Dont forget the authorization
#
$request->authorization_basic($username, $password);

# And finally make the call.
$content = $ua->request($request)->as_string;

if ($DEBUG) {
print $content;
};


if ($DEBUG) {
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
    print "+++ Daily Locked Statistics Done -
+++ \n";
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
}
sleep 60;

## ELog Cumulative Statistics Link
#
$comments_header = "<A HREF=\"http://london.ligo-
la.caltech.edu/~ops/S3/DailyStatistics/LockStatistics_".$Yest_day
."_l1_cum.html\">";
$comment_string_footer = "<P>This entry automatically elogged by
<b>ROBO SCIMON</b><BR>";
$comments_cum =  $comments_header."<b>Cumulative Locked
Statistics For $Yesterday_notime</b></A>".$comment_string_footer;
if ($DEBUG) {
   print $comments_cum."\n";
}
#
## Finally make the elog entry
# POST it with contents and values
#
my $request = POST $URL,
        Content_Type => 'multipart/form-data',
        Content =>
        [
         group => 'detector',
         task => 'makeEntry',
         log_file_date => $log_file_date,
         comments => $comments_cum,
         keywords => 'CDS',
         priority => 'normal',
         entry_author => 'cparames',
         'submit' => 'Submit Log Entry'
        ];
```

```
# Dont forget the authorization
#
$request->authorization_basic($username, $password);

# And finally make the call.
$content = $ua->request($request)->as_string;

if ($DEBUG) {
print $content;
};


if ($DEBUG) {
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
    print "+++ Cumulative Locked Statistics Done -
+++ \n";
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
}
sleep 60;

# ELog Weekly Statistics Link

if ($DAY2_STRING eq 'Sun') {
$comments_header = "<A HREF=\"http://london.ligo-
la.caltech.edu/~ops/S3/DailyStatistics/LockStatistics_".$Week_day
."_week_l1.html\">";
$comments_week =  $comments_header."<b>Weekly Locked Statistics
For $Yesterday_notime</b></A>".$comment_string_footer;
if ($DEBUG) {
   print $comments_week."\n";
}
#
# Finally make the elog entry
# POST it with contents and values
#
my $request = POST $URL,
        Content_Type => 'multipart/form-data',
        Content =>
        [
         group => 'detector',
         task => 'makeEntry',
         log_file_date => $log_file_date,
         comments => $comments_week,
         keywords => 'CDS',
         priority => 'normal',
         entry_author => 'cparames',
         'submit' => 'Submit Log Entry'
        ];
# Dont forget the authorization
#
$request->authorization_basic($username, $password);

# And finally make the call.
$content = $ua->request($request)->as_string;

if ($DEBUG) {
print $content;
};


if ($DEBUG) {
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
    print "+++ Weekly Locked Statistics Done -
+++ \n";
    print
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++ \n";
}
}
```

2) conlogheader.txt – Header file pre-pended before saving the lock statistics

```
<HTML><BODY><PRE>
```

3) conlogfooter.txt – Footer file appended before saving the lock statistics

```
</pre></BODY></HTML>
```

************************* END ****************************