

F. Chlebana
Jan 31 2007

CSL Operations Meeting

Format for CSL Operations Meeting

Current Issues

Operations Meeting Objectives

See if we can make faster progress on making the CSL system more reliable and able to run unattended

Measure progress in reducing operator intervention

Pager Report

- Summarize problems during the last week
- Has there been any new classes of problems
- Have we lost data due to CSL related problems
- Summarize what clean up procedure had to be done

We should have some measure of if we are making progress in fixing problems

Code Status

- Discuss and clarify any issues
- What new features are ready for testing

Testing Plan

- Summary of last test
- What was successful/what failed
- What is ready to test
- Can we make better use of b0csl21/b0csl30 for testing
- How will we test any new features (*Force logger switching, Generate bad events, ...*)

Monitoring Status

- Progress report on monitoring
- Status of web based monitoring
- Status of recognizing new error conditions

b0spool problem

Effects entire online cluster

Added monitoring for the b0spool/b0home areas

Reduced the load on the data area from the CSLXX machines

- Added protection in cron jobs so multiple copies are not started
- Reduced the check for disk space from 5 min to 1 hour

Suspect that the problem was caused by running a high disk io program which slowed the cron jobs so that they did not finish before the next one was called.

Tried to reproduce the problem by running the gzip job and also heavy disk io.

Saw high load on server but recovered after jobs were stopped

Logger Switching Problem

A high priority problem

Took more than 1 hour to switch

→ *During this time we lost data going to the C stream*

Problem resulted from looping when checking for an open file.

→ Stop run if we are unable to switch in less than 3 min.

→ Send warning message to error handler when we switch

→ Send error message to error handler when we fail to switch

Bad Event Handling

Redirect bad events to local disk on receiver node

Only send Bad Events to logger if they have a valid stream assigned

Run Section/Trigger Burst Problem

→ Make sure all files are added to the file manifest

→ Improve how files are split

```
1900508471  dr038e6f.001ephys    --> Section 30, 31
1901632670  dr038e6f_1.001ephys  --> Section 31, 32
1900248160  dr038e6f_2.001ephys  --> Section 32, 33
 332733475  dr038e6f_3.001ephys  --> Section 33.
1827938372  dr038e6f.0022phys    --> Section 34.
```

Can we split files between Section 31/32 for example....

Files left over in the .open area

Are they associated with starting/stopping the CSL

→ *Can we catch the stop (quite) signal and close the files gracefully*

→ *Make sure that the shift crew is using the official restart procedure*

Files Without Corresponding db

Noticed that there are “hidden” db files in the file partition area.

The file name starts with “.” and are very large (file size limit)

```
drwxrwxr-x 23 cslace cdf          4096 Jan  4 14:59 ..
-rwxrwxr--  1 cslace cdf 2147483647 Jan 19 21:43 .ar038c2d.0126phys.db
-rwxrwxr--  1 cslace cdf 2147483647 Jan 20 00:33 .ar038c2d.0281phys.db
-rwxrwxr--  1 cslace cdf 1565974528 Jan 23 17:20 .ar038d07.0410phys.db
-rwxrwxr--  1 cslace cdf  235982290 Jan 29 06:58 ar038e89.0561phys
-rwxrwxr--  1 cslace cdf 2147483647 Jan 29 07:01 .ar038e89.0561phys.db
-rwxrwxr--  1 cslace cdf    679334 Jan 30 07:00 ar038ea8.0001test
-rwxrwxr--  1 cslace cdf      290 Jan 30 07:00 ar038ea8.0001test.db
```

Looks like same pattern is being repeated in the file

```
15276246382 233097 22026706 22039353
15276245104 233097 1826463 1843429

15276246382 233097 22026706 22039353
15276245104 233097 1826463 1843429
```

Still seeing “truncated db files”

Appears less frequently

Looks like a different type of problem

Original

```
ar038d42.0001phys 996.92667103      5218  232770      969  232770  2037424
15254814721      232770      969    6044
15254814722      232770      7422   14727
...
15254814749      232770      239195 248310
15254815006      232770      3831111 3844629  <-- where did this come from
```

Regenerated

```
ar038d42.0001phys 996.92667103      5218  232770      969  232770  2037424
15254814721      232770      969    6044
15254814722      232770      7422   14727
...
15254814749      232770      239195 248310
15254814751      232770      249363 262600
...
15254814875      232770      2010293 2023728
15254814876      232770      2023984 2037424
```

Better Error Handling

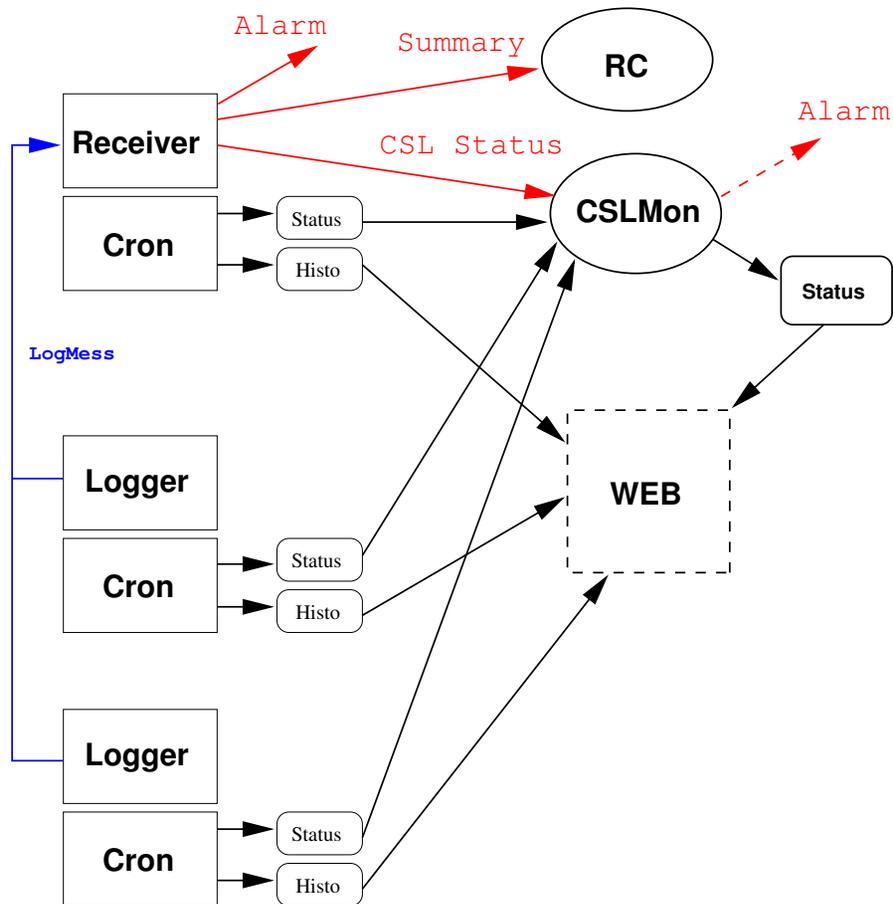
We want to make sure that errors are sent to CSLMon and relayed back to the error handling.

CSLMon has access to status information and can use this information to determine if there is a problem that requires immediate attention.

→ *Error count increasing*

→ *Logger switching/Not logging*

Flow of information



- Logger send TCP message (LogMess) to Receiver
- Receiver constructs Smart Sockets message
- Broadcast Smart Sockets message (Summary, CSL Status)
- CSLMon receives message and interprets the content
- CSLMon decides if a ERROR/WARNING/INFO message should be sent out

Web Based Monitoring

We would like to display some of important monitoring information on the web in order to be able to trouble shoot problems remotely

Will use Kurt Biery's utility to display snap shots of the display

Have CSLMon generate status page to be displayed on the web

Can we easily convert the java application to a web based display

General Cleanup

1. Make sure code is in CVS/UPS
2. Remove debug messages
3. Resolve any inconsistencies in the monitoring information
4. Average out instantaneous rates to avoid drops to 0

Make existing monitoring aware of new error conditions

1. Warning that logging has switched to new logger during a run
2. Error if no logger is available to switch to after 3 min
In this case we should force the run to stop
3. Warning that Output/Input rate has dropped
4. Warning if `CSL_NEVENTSERROR > 2`/Increasing

History Plots

1. Add queue history
2. History of Output/Input
3. History of rate to consumers

Clean Up Scripts

crontab:

```
0 10 * * * /cdf/onln/home/cslace/Script/runCorrdbfile.sh
```

Add handling of data file with no db file to cron job

Stager moves old files with no db file from:

```
/mnt/data1/logger/ace/00
```

to

```
/mnt/data1/logger/error
```

We have two different scripts that look for files in different areas.

Should consolidate into one script, use one working area and handle both the truncated db and no db file cases.

CreateDBFile.sh

```
DIRNAME=/mnt/data1/logger/ace/error
if (! -e ${DIRNAME}/${FILENAME}.db ) then
    mv ${DIRNAME}/${FILENAME} /mnt/data1/logger/ace/error/.
```

Corrdbfile.sh

```
DIRNAME=/mnt/data1/logger/error
if [ ! -e ${DIRNAME}/${FILENAME}.db ]; then
    exit
fi
```

Testing New Code

- Using version from UPS
- Better file boundary handling
- Ensure all files added to manifest
- Bad event handling
- Looping file open problem
(we lost 1 hour of data because of this)
- Fix truncated db file problem
(problem is still present)

Misc

Bad events

/mnt/data1/logger/ace/error

```
-rw-r--r--  1 cslace cdf  383928 Jan 18 22:56 00232443.021BAD
-rw-r--r--  1 cslace cdf  473536 Jan 18 22:56 00232443.022BAD
-rw-r--r--  1 cslace cdf  410140 Jan 18 22:56 00232443.023BAD
```

Collect events into single file....

Make sure runs are identified offline

Kernal/package updates ready for testing on b0csl21/b0csl30

Need to reboot b0csl21/b0csl30 and test CSL code before trying the rest...

Done with b0dau32

Disconnect/Power down

Would like to move the meeting time to Wed 1-2 pm CST to avoid a conflict with the Colloquium.