

Enduro/X Core - Feature #286

Feature # 357 (Closed): Enduro/X 7.0 master task

async logger and latent command framework (LCF)

02/10/2018 01:15 AM - Madars

Status:	Closed	Start date:	09/10/2017
Priority:	High (Code 3)	Due date:	
Assignee:		% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description <p>We shall enable new flag for debug string, which is async. This means that at init we spawn new thread which will be busy with writting data to log files. While the "clients" to logger, will just put the strings in linked lists and send for thread to write to files.</p> <p>Also at periodical intervals, async logger shall check some shared memory block where there will be rotating list of commands which shall be executed by all LCFs. The command shall be command_id, and 3x arguments (something like 32 byte chars). custom commands shall be configurable to receive callback. The xadmin shall support custom command registration.</p> <p>built in commands:</p> <ul style="list-style-type: none">- change debug levels of Enduro/X logger- re-parse cache data (requires cache thread safety on objects) <p>This will allow to change configuration of the binaries "on the fly" with out restart.</p> <p>Binaries shall keep some pointers to last executed command, and loop over to next. And what the list by timestamps (monotonic should be fine). When command is processed, thread should remember that, and also it should update the counter in command block - how much processes are executed that command. So that admin have some overview of status.</p>			
Subtasks: <p>Feature # 213: Out of the box logrotate feature</p>			
			Closed

History

#1 - 02/10/2018 02:04 PM - Madars

it shall be possible to send command to exe name or server id.
Also there shall be some header info, to point where commands start/stops?

#2 - 03/26/2018 10:22 PM - Madars

- Subject changed from *async logger and latenet command framework (LCF)* to *async logger and latent command framework (LCF)*

#3 - 07/13/2018 07:07 AM - Madars

scheduled for 5.4 a.k.a 6.0

#4 - 07/19/2018 08:07 AM - Madars

- Parent task set to #328

#5 - 08/29/2018 02:57 PM - Madars

Also it would be nice to have dump encryption facility. Something like public key loaded on disk, then in memory we generate some AES key, and then each dump consists of AES key under PUB key and dump encrypted under AES.

Also need a utility for log file decryption. The encoded dump segments must be marked specifically.

#6 - 12/03/2018 07:10 AM - Madars

- Parent task changed from #328 to #357

#7 - 03/12/2019 03:00 PM - Madars

lcf shall reload servicedb too (one that will be used for JOLT).

#8 - 03/12/2019 03:44 PM - Madars

Spinlock RW:

```
var M_readers = 0;
pthread_spinlock_t M_lock;

ndrx_rwlock_rdlock()
{
    pthread_spin_lock(&M_lock);

    M_readers++;

    pthread_spin_unlock(&M_lock);
}

ndrx_rwlock_rdunlock()
{
    pthread_spin_lock(&M_lock);

    M_readers--;

    pthread_spin_unlock(&M_lock);
}

ndrx_rwlock_wrlock();
{
    for(;;)
    {
        pthread_spin_lock(&M_lock);
        if (M_readers==0)
        {
            //If got 0 readers, then keep the master spinlock locked
            break;
        }
        pthread_spin_unlock(&M_lock);
    }
}

ndrx_rwlock_wrunlock()
{
    pthread_spin_unlock(&M_lock);
}
```

#9 - 03/22/2020 10:20 PM - Madars

We shall make remove of swait

Add asyng logging, or sync logging shall perform rw locks on the debug descriptors

#10 - 07/01/2020 06:59 PM - Madars

For log rotate seems like we need:

- Hash of file descriptor / pointer open. has sub-hash of loggers (bellow), has file name. + count of facilities. have a flag, that this pointer is used by process level loggers.
- Each file descriptor in hash, has own hash with list of logger facilities. Key by facility.

If we choose to rotate the files (re-open).

- To close some log file: find descr in hash, decrement the count, if count == 0 file shall be closed. The sub-hash entry of facility shall be removed. If involved process level files, then needs to perform full debug locking.
- To rotate all files: loop over the hash of descriptors, if has proc level logger, put global logging lock (with counters of loggers, + on exit with needs conditional variable on 0 loggers), close the pointer, open it again, add to new hash list. Loop over the sub-hash, update file pointers. Add sub-hash to new hash element. Delete from origianl top-level hash. Once all rebuilt, swap the pointers -> assign new hash table to global descriptor table.
- we could use LCF with time-stamps, to indicate to processes to which time interval command applies -> i.e. for what process age the command shall be executed. Currently we could execute the LCFs at the debug handler. Only needs to think how it would affect the logging counters, probably needs to do LCF before the logging entry/exit counters.

+ counters needs to be:

- spin-locked
- where internally there would be additional flag which would mean to switch to mutex. So that we doing log file rotate/re-open, other threads would wait on mutex.

#11 - 07/01/2020 07:09 PM - Madars

log rotate process would be:

- external script renames all log files
- calls \$ xadmin rotatelog
- external program may archive + delete (or do other actions as required). Even if process ptr still shows the old file, it would be re-open as soon as log will attempt to write by process
- thus processes would slowly re-open the logs, as they continue to work.

#12 - 07/03/2020 11:24 PM - Madars

Also might want to add lockfp=y|f

where - y lock at file pointer level

f - full locking (the debug output is fully synchronized).

#13 - 07/03/2020 11:25 PM - Madars

Also for log-rotate we might want to send notification to admin queues, so that servers process the LCF commands.

#14 - 07/11/2020 01:50 PM - Madars

In case if ndrxddebug.conf is specified file= (i.e. stderr), then we could extract the log file name from sysopts -e param from servers. And log= from client sections. The log file name could be set in environment variable. Like NDRX_PROCLLOG=... The debug init could pick up this name instead of using stderr. In that case we could do the logrotate on outside specified file name.

#18 - 02/02/2021 02:17 AM - Madars

- *Status changed from New to Resolved*

- *% Done changed from 0 to 100*

#19 - 02/28/2021 05:39 PM - Madars

- *Status changed from Resolved to Closed*

Release notes:

- Implemented LCF command framework. Published two commands:
- **xadmin lcf logrotate** - allow to reopen the log handlers
- Added **xadmin lcf** command for printing current status of the LCF commands.
- and **xadmin lcf lochg** - to change the log levels for processes with out restart.
- Added APIs for custom lcf commands: ndr_xlcf_xadmin_add() and ndr_xlcf_func_add()

See xadmin(8) and ex_env(5) man pages for more details.

with this release additional shared memory and semaphore is open.