LTTng-tools - Bug #1005

New trace can't generate in persistent memory file system

03/15/2016 11:01 PM - jia fang

Status:	Resolved	**Start date:**	03/15/2016
Priority:	Normal	**Due date:**	
Assignee:	Jonathan Rajotte Julien	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Using the new feature Recording trace data on persistent memory file systems in LTTng 2.7.

After the device restart due to system crash, we can read the crash log in the specific shm-path directory. When we want to trace the new log, the following debug log reported and the new data can't be traced to the specific shm-path.

DEBUG3 - 20:07:54.353680 [286/291]: mkdir() recursive /shm/lttng/mysession-20160302-182255/ust/uid/0/32-bit with mode 504 for uid 0 and gid 0 (in run_as_mkdir_recursive() at runas.c:468)
DEBUG1 - 20:07:54.353726 [286/291]: Using run_as worker (in run_as() at runas.c:449)
DEBUG3 - 20:07:54.354141 [286/291]: open() /shm/lttng/mysession-20160302-182255/ust/uid/0/32-bit/metadata with flags C1 mode 384 for uid 0 and gid 0 (in run_as_open() at runas.c:498)
DEBUG1 - 20:07:54.354202 [286/291]: Using run_as worker (in run_as() at runas.c:449)
PERROR - 20:07:54.354724 [286/291]: **Opening metadata file: File exists (in ust_registry_session_init() at ust-registry.c:606)**
DEBUG3 - 20:07:54.354801 [286/291]: rmdir_recursive() /shm/lttng/mysession-20160302-182255 with for uid 0 and gid 0 (in run_as_rmdir_recursive() at runas.c:524)
DEBUG1 - 20:07:54.354847 [286/291]: Using run_as worker (in run_as() at runas.c:449)
DEBUG3 - 20:07:54.355554 [287/287]: Attempting rmdir /shm/lttng/mysession-20160302-182255 (in utils_recursive_rmdir() at utils.c:1247)
DEBUG3 - 20:07:54.356905 [286/291]: Buffer registry per UID destroy with id: 0, ABI: 32, uid: 0 (in buffer_reg_uid_destroy() at buffer-registry.c:641)

For detail info, please see my attachment.

Related issues:

Copied from Common Trace Format - Bug #1004: New trace can't generate in pers...	**Invalid**	**03/15/2016**

History

#1 - 03/15/2016 11:02 PM - jia fang

- Copied from Bug #1004: New trace can't generate in persistent memory file system added

#2 - 03/18/2016 05:32 PM - Jonathan Rajotte Julien

- Status changed from New to Feedback

- Assignee set to Jonathan Rajotte Julien

- Priority changed from High to Normal

Hi Jia,

Does the file exists as the error suggests ?

Do you use lttng load and a xml session descriptor?

The main reason to use shm is minimize data loss so not squashing existing files make sense since we do not know if the user have collected them. Depending on your use case there is multiples solutions.

Cheers

#3 - 03/21/2016 05:49 AM - jia fang

Hi,

yes, we use lttng load and xml session descriptor.

My use case is:

The system crashed and then it reboot and set up, after that, I run my application with the same shm-path.
I want to the new log of my application can be record into my shm-path file.

I try to open shm-path file with O_APPEND or O_TRUNC, it seems like that my application can overwrite the system crash log.

Could you please give some suggestion ?
Or I have to save and delete my crash log manually, and then run my application ?

Thanks

#4 - 03/21/2016 12:05 PM - Jonathan Rajotte Julien

jia fang wrote:

> Hi,
>
> yes, we use lttng load and xml session descriptor.

So technically you control were the shm path is located and the trace output and these locations are static.

Could you attach the actual xml files for the load command (make sure to replace any confidential informations by fake ones)?

A list of the used commands would be helpful to.

> My use case is:
> The system crashed and then it reboot and set up, after that, I run my application with the same shm-path.
> I want to the new log of my application can be record into my shm-path file.

If you have a set-up phase I would recommend checking for files presence in the shm path and tar it before starting lttng-sessiond/using lttng. All this automatically. Make sure to also delete the files.

> I try to open shm-path file with O_APPEND or O_TRUNC, it seems like that my application can overwrite the system crash log.

What is your procedure on a system crash ? Reboot and ignore the shm files or save the shm files for further extraction/analysis?

> Could you please give some suggestion ?
> Or I have to save and delete my crash log manually, and then run my application ?

I'm sure you could do this easily automatically on boot.

> Thanks

#5 - 03/21/2016 10:25 PM - jia fang

Jonathan Rajotte Julien wrote:

> jia fang wrote:
>
> > Hi,
> >
> > yes, we use lttng load and xml session descriptor.
>
> So technically you control were the shm path is located and the trace output and these locations are static.
>
> Could you attach the actual xml files for the load command (make sure to replace any confidential informations by fake ones)?
>
> A list of the used commands would be helpful to.

Attachment is my xml files

> > My use case is:
> > The system crashed and then it reboot and set up, after that, I run my application with the same shm-path.
> > I want to the new log of my application can be record into my shm-path file.
>
> If you have a set-up phase I would recommend checking for files presence in the shm path and tar it before starting lttng-sessiond/using lttng. All this automatically. Make sure to also delete the files.

So the easy way is to save and delete the files during the set-up phase before running application :)

> I try to open shm-path file with O_APPEND or O_TRUNC, it seems like that my application can overwrite the system crash log.

What is your procedure on a system crash ? Reboot and ignore the shm files or save the shm files for further extraction/analysis?

My crash is "echo c > /proc/sysrq-trigger", and then my system will reboot automatically.
I want to reboot and "cp" my shm files but not "mv" (that is to say, I did not delete the shm files during set-up). After that, I run my application and then new log can be written into my shm files.
So I choose to use O_APPEND or O_TRUNC to open my shm files.

BTW, I found that even O_APPEND will not append new log to shm files. Both O_APPEND and O_TRUNC overwrite the shm files.

> Could you please give some suggestion ?
> Or I have to save and delete my crash log manually, and then run my application ?

I'm sure you could do this easily automatically on boot.

> Thanks

#6 - 03/21/2016 10:27 PM - jia fang

- File mysession.lttng added

#7 - 03/24/2016 12:34 PM - Jonathan Rajotte Julien

jia fang wrote:

> Jonathan Rajotte Julien wrote:
>
> > jia fang wrote:
> >
> > > Hi,
> > >
> > > yes, we use lttng load and xml session descriptor.
> >
> > So technically you control were the shm path is located and the trace output and these locations are static.
> >
> > Could you attach the actual xml files for the load command (make sure to replace any confidential informations by fake ones)?
> >
> > A list of the used commands would be helpful to.
>
> Attachment is my xml files

Looks good to me.

> > My use case is:
> > The system crashed and then it reboot and set up, after that, I run my application with the same shm-path.
> > I want to the new log of my application can be record into my shm-path file.
>
> If you have a set-up phase I would recommend checking for files presence in the shm path and tar it before starting lttng-sessiond/using lttng. All this automatically. Make sure to also delete the files.

So the easy way is to save and delete the files during the set-up phase before running application :)

Yes since lttng will not squash existing shm files **by design** since it cannot know if the user have extracted/saved them.

> I try to open shm-path file with O_APPEND or O_TRUNC, it seems like that my application can overwrite the system crash log.

What is your procedure on a system crash ? Reboot and ignore the shm files or save the shm files for further extraction/analysis?

> My crash is "echo c > /proc/sysrq-trigger", and then my system will reboot automatically.
> I want to reboot and "cp" my shm files but not "mv" (that is to say, I did not delete the shm files during set-up). After that, I run my application and then new log can be written into my shm files.
> So I choose to use O_APPEND or O_TRUNC to open my shm files.

Why not 'mv'?

Where did you use O_APPEND/O_TRUNC ?

The principal scenario for using shm path would be:
1) The target crash
2) On set-up you have a script that look for orphan lttng files (in the shm path) and move them to a safe place for further analysis
3) Start lttng and tracing

Cheers

#8 - 03/29/2016 05:48 AM - jia fang

Jonathan Rajotte Julien wrote:

> jia fang wrote:
>
> > Jonathan Rajotte Julien wrote:
> >
> > > jia fang wrote:
> > >
> > > > Hi,
> > > >
> > > > yes, we use lttng load and xml session descriptor.
> > >
> > > So technically you control were the shm path is located and the trace output and these locations are static.
> > >
> > > Could you attach the actual xml files for the load command (make sure to replace any confidential informations by fake ones)?
> > >
> > > A list of the used commands would be helpful to.
> >
> > Attachment is my xml files
>
> Looks good to me.
>
> > > My use case is:
> > > The system crashed and then it reboot and set up, after that, I run my application with the same shm-path.
> > > I want to the new log of my application can be record into my shm-path file.
> >
> > If you have a set-up phase I would recommend checking for files presence in the shm path and tar it before starting lttng-sessiond/using lttng. All this automatically. Make sure to also delete the files.
>
> So the easy way is to save and delete the files during the set-up phase before running application :)

Yes since lttng will not squash existing shm files **by design** since it cannot know if the user have extracted/saved them.

> > I try to open shm-path file with O_APPEND or O_TRUNC, it seems like that my application can overwrite the system crash log.
> >
> > What is your procedure on a system crash ? Reboot and ignore the shm files or save the shm files for further extraction/analysis?
>
> My crash is "echo c > /proc/sysrq-trigger", and then my system will reboot automatically.
> I want to reboot and "cp" my shm files but not "mv" (that is to say, I did not delete the shm files during set-up). After that, I run my application and then new log can be written into my shm files.
> So I choose to use O_APPEND or O_TRUNC to open my shm files.

Why not 'mv'?

Because at that time I did not know I need to 'mv' :)

> Where did you use O_APPEND/O_TRUNC ?

run_as_open(session->metadata_path...) in lttng-tools/src/bin/lttng-sessiond/ust-registry.c and run_as_open(session->metadata_path...) in lttng-tools/src/common/ust-consumer/ust-consumer.c

Do I misunderstand something ?

BRs

> The principal scenario for using shm path would be:
> 1) The target crash
> 2) On set-up you have a script that look for orphan lttng files (in the shm path) and move them to a safe place for further analysis
> 3) Start lttng and tracing
>
> Cheers

#9 - 03/29/2016 04:04 PM - Jonathan Rajotte Julien

Good.

Regarding the O_APPEND/O_TRUNC I would recommend against doing this change for the sake of crash data persistence.

With the change made to the set-up phase are you satisfied with the lttng shm feature and the lttng-crash utility ?

Cheers

#10 - 03/30/2016 09:55 PM - jia fang

Jonathan Rajotte Julien wrote:

> Good.
>
> Regarding the O_APPEND/O_TRUNC I would recommend against doing this change for the sake of crash data persistence.

OK, Thank you so much.

> With the change made to the set-up phase are you satisfied with the lttng shm feature and the lttng-crash utility ?

We will change the set-up phase and other features are fine for us. If any problem, we will update.
Thanks again for quick reply :)

> Cheers

#11 - 04/07/2016 04:28 PM - Jonathan Rajotte Julien

- Status changed from Feedback to Resolved

Files

lttng debug log	7.37 KB	03/15/2016	jia fang
lttng debug log	7.37 KB	03/16/2016	jia fang
mysession.lttng	1.76 KB	03/22/2016	jia fang