Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Bug #965

Audacious often crashes or freezes when a new cue file is played almost immediately after the end of the current playlist is reached

April 13, 2020 12:19 - Igor Kushnir

Status:	Closed	**Start date:**	
Priority:	Minor	**Due date:**	
Assignee:		**% Done:**	100%
Category:	plugins/alsa	**Estimated time:**	0.00 hour
Target version:	4.0.3		
Affects version:	4.0.1		

Description

Steps to reproduce:
1. Enable Song Change Audacious plugin and configure it as follows. Command to run at the end of the playlist: bash -c "sleep 0.0005 && audacious -p -E '/home/path/to/some filename.cue'"
2. Disable Repeat and Shuffle options in Audacious to let the configured Song Change command run.
3. Navigate very close (a second or two is fine) to the end of the last song in the current playlist (e.g. by double clicking on the last track and then clicking near the end of the progress bar).
4. If Audacious doesn't crash or freeze, repeat the previous step a few times. If Audacious does not crash or freeze after a few repetitions, change the sleep interval. Sleep interval 0.05 sometimes works better for `audacious -V -V`; for `audacious` without arguments 0.01, 0.005, 0.001, 0.0005 often allow to reproduce the bug after a few tries.

Notes:
1) For some reason on different Audacious launches the time interval that results in a crash or freeze may differ. For example, I could reproduce the freeze a few times in a row with the interval 0.0005, but then it just stopped freezing until I changed the interval to 0.005. I don't know if this is a real difference or just random luck.
2) I can reproduce the bug with my system Audacious (the versions of audacious and audacious-plugins Manjaro packages are 4.0.1-1 - exactly the same as in Arch, built by Evangelos Foutras) and with Audacious built from sources - from master and 4.0.x branches.
3) When I test the system version of Audacious, it usually crashes; the version built from sources usually freezes. This is due to different build flags I think.
4) When Audacious freezes, `killall audacious` is not enough, only `killall -9 audacious` kills the process.
5) I have encountered different stack traces while reproducing the bug. Attached those captured by coredumpctl to this bug.
6) I am affected by this issue because I am using my own helper application that instructs Audacious what to play next. The application is notified by the Song Change plugin. It reacts within one second and launches the "audacious -p -E '/home/path/to/some filename.cue'" command. Sometimes, when my application reacts early enough I suppose, Audacious crashes. My current workaround is to delay my application by a guaranteed one second.
7) I have been using my application in this way with different Audacious 3.x versions for years without crashes or freezes. This definitely started a few days ago when Manjaro audacious* packages were upgraded from 3.10.1 to 4.0.1. So this must be either a 4.x or a Qt interface regression.

History

#1 - April 13, 2020 14:14 - John Lindgren

Can you reinstall Audacious with debug symbols? The backtraces are not very useful otherwise.

#2 - April 13, 2020 14:39 - Igor Kushnir

I have already installed audacious and audacious-plugins built with debug symbols into /usr/local, but as I mentioned in the bug description, the version I built myself tends to freeze rather than crash and so doesn't produce a core dump or a backtrace. I've been able to view a frame stack in KDevelop a few times (it didn't look to me like anything easy to fix), but usually I just can't get anything at all out of a frozen Audacious process. If you can not reproduce this bug, I'll try to generate better backtraces.

#3 - April 13, 2020 14:47 - John Lindgren

I get a warning from the ALSA plugin, but no crash:

```
ERROR alsa.cc:255 [get_delay_locked]: snd_pcm_recover failed: File descriptor in bad state.
```

#4 - April 13, 2020 15:15 - John Lindgren

Fixed the ALSA warning:
https://github.com/audacious-media-player/audacious-plugins/commit/c1a594bcc335acd9eec286b3c65b8fecc91e3d20

It looks like you are using PulseAudio. Can you change the output plugin to ALSA and see if that makes the crash go away?

#5 - April 13, 2020 15:31 - Igor Kushnir

I also had similar warnings in terminal and non-blocking error messages in the UI. Had to experiment with the sleep delay to make Audacious crash or freeze.

I have tested flac files directly instead of cue files that reference them, but never got a freeze or a crash without a cue intermediary. The two cue files I have reproduced the crash with have 17 and 22 tracks, they reference flac files of size 375 and 245 MiB respectively. This bug is not likely to be specific to the two cue files, because I chose them at random from my collection, and both allowed me to reproduce the crash. But the underlying file format (flac), file size (more than 200 MiB) and the significant number of tracks could make a difference.

#6 - April 13, 2020 15:44 - Igor Kushnir

John Lindgren wrote:

> Fixed the ALSA warning:
> https://github.com/audacious-media-player/audacious-plugins/commit/c1a594bcc335acd9eec286b3c65b8fecc91e3d20
>
> It looks like you are using PulseAudio. Can you change the output plugin to ALSA and see if that makes the crash go away?

In fact, Output plugin was set to "ALSA Output" in my Audacious settings. When I set it to "PulseAudio Output", I can't reproduce the crash.

Just got the following alsa warning while trying to reproduce the crash twice, then a freeze:

```
$ audacious
ERROR alsa.cc:255 [get_delay_locked]: snd_pcm_recover failed: Input/output error.
ERROR alsa.cc:255 [get_delay_locked]: snd_pcm_recover failed: Input/output error.
Killed
```

I'll test your commit now.

#7 - April 13, 2020 16:02 - Igor Kushnir

I pulled your commit, rebuilt and reinstalled audacious-plugins (but not audacious). Audacious still freezes with some sleep intervals and ALSA Output. Didn't get any alsa warnings after several attempts though.

#8 - April 13, 2020 16:27 - John Lindgren

It's almost certainly timing-related. You're hitting Audacious with a request to play a new track right at the end of the last one, as the audio buffer is draining. This is a transition state for the internals of the program, and some parts (e.g. decoding) are already done with the track while others are not. I made some changes in 4.0.x to make the UI more responsive in this state. A Git bisect would probably lead you to this commit[1], but the bug isn't necessarily in that commit itself; more likely, it exposed some underlying issues (like the one in the ALSA plugin that I just fixed).

Some backtraces with debug symbols would be very helpful. I would suggest simply taking the Arch Linux PKGBUILDs and rebuilding them with the "debug" option enabled (see the makepkg manpage[2]).

[1] https://github.com/audacious-media-player/audacious/commit/cd50db1079e0d0556c1bc5a700b91a9b27602a3c

[2] https://www.archlinux.org/pacman/makepkg.conf.5.html

#9 - April 13, 2020 18:58 - Igor Kushnir

- File audacious-4.0.1-1-debug-0.1s.backtrace added

- File audacious-git-master-debug-0.2s.backtrace added

- File audacious-git-master-debug-0.2s-2.backtrace added

- File audacious-git-master-debug-0.2s-freeze.backtrace added

- File audacious-git-master-debug-0.2s-freeze-2.backtrace added

- File audacious-git-master-debug-0.2s-freeze-3.backtrace added

- File audacious-git-master-debug-0.2s-3.backtrace added

Now I suspect there are at least two bugs here. Crashes and freezes are not the same issues. Plus I got two different signals and backtraces for crashes. Reproducing a crash is more difficult than a freeze. I increased the sleep interval to 0.1, then to 0.2 to get a crash. In order to reproduce a crash I had to:

1. Start Audacious with command: *audacious -p*
2. Navigate to at least 10 seconds before the end of the last song.
3. Switch to another maximized application and leave Audacious to finish playback in the background.
4. Repeat the previous two steps until Audacious crashes or freezes. If it freezes, execute *"killall -9 audacious"* and go to step 1.

I have also found a way to extract a backtrace from a frozen application:

```
gdb -batch -ex "thread apply all bt" -ex "quit" -p 74310 > /path/to/backtrace
```

Attaching both crash and frozen backtraces with debug symbols. One backtrace is from Audacious 4.0.1 (Arch package with debug info), 0.1s delay; all others - from the current git master (both audacious and audacious-plugins), 0.2s delay.

Note the presence of nouveau_dri.so in the backtraces. There are known thread safety bugs in nouveau driver, so some of the crashes or freezes might be caused by it. Although I recall such crashes were mostly in Qt WebEngine code, which Audacious does not use.

#10 - April 13, 2020 19:02 - Igor Kushnir

Forgot to mention that I got the following output from crashing Audacious in terminal:

```
$ audacious -p
terminate called after throwing an instance of 'std::system_error'
  what():  No such process
Aborted (core dumped)
```

#11 - April 13, 2020 19:37 - John Lindgren

I am inclined to say that you have a libpthread, compiler, or CPU/firmware bug going on here. There are things in the backtraces that simply make no sense (std::terminate called when joining joinable() threads, deadlocks that are logically impossible, etc.)

I cannot help you further with this, sorry. It would take significant time, and access to the real hardware, to figure out what is really going on.

#12 - April 20, 2020 15:03 - Igor Kushnir

- File audacious-git-master-freeze-with-debug-output.backtrace added

- File audacious-git-master-freeze-with-debug-output.stderr added

- File 0001-Detect-pump_stop-deadlock.patch added

I have been able to narrow down the freeze, but not the crash yet. Attached my deadlock-detecting patch, stderr output and the backtrace of the frozen process.

As I understand, there is a race condition between *drain()* , *flush()* and *pump()*. In this case *drain()* and *flush()* both broadcast and attempt to join *pump_thread*. One of the broadcasts is lost, so only *drain()* succeeds, while *flush()* remains waiting indefinitely.

#13 - April 20, 2020 16:16 - Igor Kushnir

I have spent some time debugging the crash without success. Then switched to debugging the freeze and narrowed it down in no time. So I'm going to suspend debugging the crash until the freezing is fixed. There is hope that the fix will cover both issues. After all I can't reproduce the crash with PulseAudio output plugin. Perhaps this race condition corrupts pthread's private data, which then causes the crash. From *man 3* pthread_join:

> If multiple threads simultaneously try to join with the same thread, the results are undefined.

#14 - April 20, 2020 17:39 - John Lindgren

Aha, that's some very good detective work. Thank you. I'll need to put some thought into how to solve this.

#15 - April 20, 2020 19:15 - John Lindgren

See if this helps:

https://github.com/audacious-media-player/audacious-plugins/commit/0dac5d9a451700ffbbbd85427dc2d3661dc85bdb

#16 - April 21, 2020 08:25 - Igor Kushnir

I installed git master versions of audacious and audacious-plugins and tried to reproduce the crash or the freeze maybe 30-40 times with different sleep delays. None happened. Thanks for the complete fix.

However during my testing with sleep delay 0.1 two times the following error appeared in the UI and in *audacious -p* output:

```
ERROR alsa.cc:253 [get_delay_locked]: snd_pcm_recover failed: Input/output error.
```

Also, as used to happen during my earlier tests, there were the following string leak warnings in the output after I exited Audacious:

```
$ audacious -p
ERROR alsa.cc:253 [get_delay_locked]: snd_pcm_recover failed: Input/output error.
ERROR alsa.cc:253 [get_delay_locked]: snd_pcm_recover failed: Input/output error.
WARNING strpool.cc:181 [operator()]: String leaked: $PERFORMER - $TITLE - ${TRACK 01 TITLE}
WARNING strpool.cc:181 [operator()]: String leaked: $TITLE
WARNING strpool.cc:181 [operator()]: String leaked: ${PATH TO THE DIRECTORY CONTAINING THE CUE FILE}
WARNING strpool.cc:181 [operator()]: String leaked: Free Lossless Audio Codec (FLAC)
WARNING strpool.cc:181 [operator()]: String leaked: file://${URL-ENCODED ABSOLUTE PATH}.flac
WARNING strpool.cc:181 [operator()]: String leaked: $PERFORMER
WARNING strpool.cc:181 [operator()]: String leaked: $GENRE
WARNING strpool.cc:181 [operator()]: String leaked: ${TRACK 01 TITLE}
WARNING strpool.cc:181 [operator()]: String leaked: cue
WARNING strpool.cc:181 [operator()]: String leaked: lossless
WARNING strpool.cc:181 [operator()]: String leaked: file://${URL-ENCODED ABSOLUTE PATH}.cue?1
WARNING runtime.cc:364 [aud_leak_check]: Bytes allocated at exit: 7360
```

I replaced the actual strings with $VARIABLE, ${VARIABLE} as specified in the cue file that was in the tested *bash -c "sleep 0.0005 && audacious -p -E '/home/path/to/some filename.cue'"* command.
Note that these warnings happened often before, even in the absence of alsa errors.

#17 - April 21, 2020 13:24 - John Lindgren

- % Done changed from 0 to 100

- Status changed from New to Closed

- Category set to plugins/alsa

#18 - April 21, 2020 13:25 - John Lindgren

- Target version set to 4.0.3

I'll go ahead and close this, and if you can narrow down the conditions that lead to the leak warnings, please open a new report. Thanks for testing.

Files

coredumpctl info audacious.txt	93.9 KB	April 13, 2020	Igor Kushnir
audacious-git-master-debug-0.2s-3.backtrace	10.3 KB	April 13, 2020	Igor Kushnir
audacious-git-master-debug-0.2s-freeze-3.backtrace	9.63 KB	April 13, 2020	Igor Kushnir
audacious-git-master-debug-0.2s-freeze-2.backtrace	9.63 KB	April 13, 2020	Igor Kushnir
audacious-git-master-debug-0.2s-2.backtrace	10.4 KB	April 13, 2020	Igor Kushnir
audacious-git-master-debug-0.2s.backtrace	10.3 KB	April 13, 2020	Igor Kushnir
audacious-git-master-debug-0.2s-freeze.backtrace	9.63 KB	April 13, 2020	Igor Kushnir
audacious-4.0.1-1-debug-0.1s.backtrace	7.24 KB	April 13, 2020	Igor Kushnir
0001-Detect-pump_stop-deadlock.patch	1.95 KB	April 20, 2020	Igor Kushnir
audacious-git-master-freeze-with-debug-output.backtrace	9.76 KB	April 20, 2020	Igor Kushnir
audacious-git-master-freeze-with-debug-output.stderr	461 Bytes	April 20, 2020	Igor Kushnir