
Subject: Re: [RFC][PATCH][cryo] Save/restore state of unnamed pipes
Posted by [Sukadev Bhattiprolu](#) on Thu, 19 Jun 2008 07:59:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Matt Helsley [matthlhc@us.ibm.com] wrote:

<snip>

```
| > | I don't see anything in the pipe man page, at least, that suggests we  
| > | can safely assume pipefds[0] < pipefds[1].  
| > |  
| > | The solution could be to use "trampoline" fds. Suppose last_fd is the  
| > | largest fd that exists in the final checkpointed/restarting application.  
| > | We could do (Skipping the PT_FUNC "notation" for clarity):  
| > |  
| > |  
| > |     pipe(pipefds); /* returns 5 and 4 in elements 0 and 1 */  
| > |     /* use fds after last_fd as trampolines for fds we want to create */  
| > |     dup2(pipefds[0], last_fd + 1);  
| > |     dup2(pipefds[1], last_fd + 2);  
| > |     close(pipefds[0]);  
| > |     close(pipefds[1]);  
| > |     dup2(last_fd + 1, <orig pipefd[0]>);  
| > |     dup2(last_fd + 2, <orig pipefd[1]>);  
| > |     close(last_fd + 1);  
| > |     close(last_fd + 2);  
| > |  
| > |  
| > | Which is alot more code but should work no matter which fds we get back  
| > | from pipe(). Of course this assumes the checkpointed application hasn't  
| > | used all of its fds. :(  
| > |
```

It appears that this last_fd approach will fit in easier with current design of cryo (where we process one or two fds at a time and don't have the src_fds and dest_fds handy).

BTW, we should be able to accomplish the above with a single-unused fd right (i.e no need for last_fd+2) ?

```
| >  
| > This sounds like a good idea too, but we could use any fd that has not  
| > yet been used in the restart-process right ? It would break if all fds  
|  
| Yes, but we don't know which fd is available unless we allocate it  
| without dup2().
```

Right. I was thinking we could find that out at the time of checkpoint (a brute-force `fstat(i, &statbuf)` for `i = 0..n` or something more efficient).

Well just thought of another approach.

Basically, we have a temporary need for an unused fd for use as a trampoline. So, why not 'set-aside' an fd for that purpose and after all other fds have been created, go back and create this fd ?

i.e lets say the first regular file we open is associated with 'fd = 3'. We save away the 'fdinfo' for 3 say in a global variable and `close(3)`. Now use 'fd = 3' in place of `last_fd+1` above.

Once all fds have been setup correctly, go back and set up `fd = 3` using the saved fdinfo (this would be a simple open of the file followed by seek and maybe an `fcntl`).

This would work even if the application was using all its fds ?

If we do need both `last_fd+1` and `last_fd+2`, we would have to set aside two regular files.

Hmm, would it work even if an application uses all (1024) its fds for pipes :-), but just a thought at this point.

Suka

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
