Re: [PATCH 09/13] aio: add support for async openat()

[Posted January 12, 2016 by corbet]

From: Linus Torvalds <torvalds-AT-linux-foundation.org>
To: Benjamin LaHaise <bcrl-AT-kvack.org>, Ingo Molnar <mingo-AT-kernel.org>
Subject: Re: [PATCH 09/13] aio: add support for async openat()
Date: Mon, 11 Jan 2016 16:22:28 -0800
Message-ID: <CA+55aFw8j_3Vkb=HVoMwWTPD=5ve8RpNZeL31CcKQZ+HRSbfTA@mail.gmail.com>
Cc: linux-aio-AT-kvack.org, linux-fsdevel <linux-fsdevel-AT-vger.kernel.org>, Linux Kernel Mailing List <linux-kernel-AT-vger.kernel.org>, Linux API <linux-api-AT-vger.kernel.org>, linux-mm <linux-mm-AT-kvack.org>, Alexander Viro <viro-AT-zeniv.linux.org.uk>, Andrew Morton <akpm-AT-linux-foundation.org>
Archive-link: Article, Thread

```
On Mon, Jan 11, 2016 at 2:07 PM, Benjamin LaHaise <bcrl@kvack.org> wrote:
> Another blocking operation used by applications that want aio
> functionality is that of opening files that are not resident in memory.
> Using the thread based aio helper, add support for IOCB_CMD_OPENAT.

So I think this is ridiculously ugly.

AIO is a horrible ad-hoc design, with the main excuse being "other,
less gifted people, made that design, and we are implementing it for
compatibility because database people - who seldom have any shred of
taste - actually use it".

But AIO was always really really ugly.

Now you introduce the notion of doing almost arbitrary system calls
asynchronously in threads, but then you use that ass-backwards nasty
interface to do so.

Why?

If you want to do arbitrary asynchronous system calls, just *do* it.
But do _that_, not "let's extend this horrible interface in arbitrary
random ways one special system call at a time".

In other words, why is the interface not simply: "do arbitrary system
call X with arguments A, B, C, D asynchronously using a kernel
thread".

That's something that a lot of people might use. In fact, if they can
avoid the nasty AIO interface, maybe they'll even use it for things
like read() and write().

So I really think it would be a nice thing to allow some kind of
arbitrary "queue up asynchronous system call" model.

But I do not think the AIO model should be the model used for that,
even if I think there might be some shared infrastructure.

So I would seriously suggest:

 - how about we add a true "asynchronous system call" interface

 - make it be a list of system calls with a futex completion for each
list entry, so that you can easily wait for the end result that way.

 - maybe (and this is where it gets really iffy) you could even pass
in the result of one system call to the next, so that you can do
things like

       fd = openat(..)
       ret = read(fd, ..)

   asynchronously and then just wait for the read() to complete.

and let us *not* tie this to the aio interface.

In fact, if we do it well, we can go the other way, and try to
implement the nasty AIO interface on top of the generic "just do
things asynchronously".

And I actually think many of your kernel thread parts are good for a
generic implementation. That whole "AIO_THREAD_NEED_CRED" etc logic
all makes sense, although I do suspect you could just make it
unconditional. The cost of a few atomics shouldn't be excessive when
we're talking "use a thread to do op X".

What do you think? Do you think it might be possible to aim for a
generic "do system call asynchronously" model instead?

I'm adding Ingo the to cc, because I think Ingo had a "run this list
of system calls" patch at one point - in order to avoid system call
overhead. I don't think that was very interesting (because system call
overhead is seldom all that noticeable for any interesting system
calls), but with the "let's do the list asynchronously" addition it
might be much more intriguing. Ingo, do I remember correctly that it
was you? I might be confused about who wrote that patch, and I can't
find it now.

               Linus

--
To unsubscribe, send a message with 'unsubscribe linux-mm' in
the body to majordomo@kvack.org.  For more info on Linux MM,
see: http://www.linux-mm.org/ .
Don't email: <a href=mailto:"dont@kvack.org"> email@kvack.org </a>
```