Subject: Re: [PATCH v8 0/3] cgroups: implement moving a threadgroup's threads atomically with cgroup.procs
Posted by Paul Menage on Mon, 14 Feb 2011 06:12:51 GMT

View Forum Message <> Reply to Message

On Wed, Feb 9, 2011 at 3:10 PM, Andrew Morton <akpm@linux-foundation.org> wrote:
> On Mon, 7 Feb 2011 20:35:42 -0500
> Ben Blum <bblum@andrew.cmu.edu> wrote:
>
>> On Sun, Dec 26, 2010 at 07:09:19AM -0500, Ben Blum wrote:
>> > On Fri, Dec 24, 2010 at 03:22:26AM -0500, Ben Blum wrote:
>> > > On Wed, Aug 11, 2010 at 01:46:04AM -0400, Ben Blum wrote:
>> > > > On Fri, Jul 30, 2010 at 07:56:49PM -0400, Ben Blum wrote:
>> > > > > This patch series is a revision of http://lkml.org/lkml/2010/6/25/11 .
>> > > > >
>> > > > > This patch series implements a write function for the 'cgroup.procs'
>> > > > > per-cgroup file, which enables atomic movement of multithreaded
>> > > > > applications between cgroups. Writing the thread-ID of any thread in a
>> > > > > threadgroup to a cgroup's procs file causes all threads in the group to
>> > > > > be moved to that cgroup safely with respect to threads forking/exiting.
>> > > > > (Possible usage scenario: If running a multithreaded build system that
>> > > > > sucks up system resources, this lets you restrict it all at once into a
>> > > > > new cgroup to keep it under control.)
>> > > > >
>
> The above six month old text is the best (and almost the only)
> explanation of the rationale for the entire patch series. Is
> it still correct and complete?
>

It's still correct, but I'm sure we could come up with a more detailed
justification if necessary.

Paul