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 06:12:19 GMT
View Forum Message <> Reply to Message

On Wed, Feb 9, 2011 at 5:02 PM, KAMEZAWA Hiroyuki
<kamezawa.hiroyu@jp.fujitsu.com> wrote:
>
> So, I think it's ok to have 'procs' interface for cgroup if
> overhead/impact of patch is not heavy.
>

Agreed - it's definitely an operation that comes up as either confusing or annoying for users, depending on whether or not they understand how threads and cgroups interact. (We've been getting people wanting to do this internally at Google, and I'm guessing that we're one of the bigger users of cgroups.)

In theory it's something that could be handled in userspace, in one of two ways:

- repeatedly scan the old cgroup's tasks file and sweep any threads from the given process into the destination cgroup, until you complete a clean sweep finding none. (Possibly even this is racy if a thread is being slow to fork)

- use a process event notifier to catch thread fork events and keep track of any newly created threads that appear after your first sweep of threads, and be prepared to handle them for some reasonable length of time (tens of milliseconds?) after the last thread has been apparently moved.

(The alternative approach, of course, is to give up and never try to move a process into a cgroup except right when you're in the middle of forking it, before the exec(), when you know that it has only a single thread and you're in control of it.)

These are both painful procedures, compared to the very simple approach of letting the kernel move the entire process atomically.

It's true that it's a pretty heavyweight operation, but that weight is only paid when you actually use it on a very large process (and which would be even more expensive to do in userspace). For the rest of the kernel, it's just an extra read lock in the fork path on a semaphore in a structure that's pretty much guaranteed to be in cache.

Paul

_____
Containers mailing list