Subject: Re: [RFC][PATCH] allow "unlimited" limit value.
Posted by David Rientjes on Tue, 25 Sep 2007 19:30:39 GMT
View Forum Message <> Reply to Message

On Wed, 26 Sep 2007, Balbir Singh wrote:

> Yes, I prefer 0 as well and had that in a series in the Lost World
> of my earlier memory/RSS controller patches. I feel now that 0 is
> a bit confusing, we don't use 0 to mean unlimited, unless we
> treat the memory.limit_in_bytes value as boolean. 0 is false,
> meaning there is no limit, > 0 is true, which means the limit
> is set and the value is specified to the value read out.
>

I think any user who attaches a task that is still running to cgroup that
has memory.limit_in_bytes specified as 0 will realize quickly that it's
not doing anything to limit memory. 0 is the best choice for denoting
unlimited memory limits.

```
> > diff --git a/kernel/res_counter.c b/kernel/res_counter.c
> > --- a/kernel/res_counter.c
> > +++ b/kernel/res_counter.c
> > @@ -16,12 +16,15 @@
> >  void res_counter_init(struct res_counter *counter)
> >  {
> >   spin_lock_init(&counter->lock);
> > - counter->limit = (unsigned long)LONG_MAX;
>
> So, we create all containers with infinite limit?
>
```

Yeah, since you kzalloc'd the struct mem_cgroup, the struct res_counter
will also be zero'd and inherently have a limit of 0. It's far better
than any arbitrary value you're going to give them, unless they inherit
the value of their parent.

```
> >  }
> >
> >  int res_counter_charge_locked(struct res_counter *counter, unsigned long val)
> >  {
> > - if (counter->usage + val > counter->limit) {
> > + /*
> > +  * If 'memory.limit' is set to 0, there is no charge to this
>
> nit pick, should be memory.limit_in_bytes
>
```

This is from a month ago, I'm assuming more has changed than just the name here :)

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