Subject: Re: Q: Do systems using containers user more process ids?
Posted by ebiederm on Tue, 15 Aug 2006 18:32:48 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

>>>We have not seen any degradation here in real cases,
>>>but probably you are right and pid hash can be allocated taking into account
>>>physical memory as it is done for TCP/ip/other hashes?
>>
>>
>> It is but it is currently capped at 4K entries.
>> With 4K entries and 32K pids our worst case is usage is a hash chain
>> 9 entries long. At 4M pids our hash chains are 1000 entries long, which
>> sucks.
> 4M pids are almost unreal in production systems.
> (only if you spawn these tasks to make them sleep forever :)))).
> we usually have no more than 20,000 tasks (which is 200VEs with 100 tasks in
> each)

Ok. That is a reasonable upper bound. I need to look and see what a heavily
loaded sever looks like.

>> The practical question is if systems using containers are using noticeably
>> more pids than anyone else. So far the responses I have gotten indicate
>> that users aren't. So at least until we descend into multi-core madness
>> it sounds like the current structures are fine, but it might be worth moving
>> the cap on the number of pid hash table entries at some point in the future.
> containers are using noticeably more pids, I think it is not a doubt...
> the question is whether it is worth doing something here _now_...

True. The break even point in terms of cache misses between a hash chain
and a binary tree is between 8 and 16 levels deep. Currently it looks like
we are close to that point but not over on a heavily loaded system.

So until we cross that line it probably doesn't make sense to change
the implementation, unless it makes multiple pid namespaces easier to
implement, and it doesn't look like it will significantly help that
case.

Eric