Subject: Re: [PATCH 4/7] UBC: syscalls (user interface)
Posted by Andrey Savochkin on Fri, 18 Aug 2006 08:08:09 GMT
View Forum Message <> Reply to Message

Hi,

On Thu, Aug 17, 2006 at 08:40:33AM -0700, Andrew Morton wrote:
> On Thu, 17 Aug 2006 16:13:30 +0400
> Kirill Korotaev <dev@sw.ru> wrote:
>
> > > I was more thinking about (for example) user land physical memory limit
> > > for that bean counter. If the limits are going down, then the system
> > > call should try to flush out page cache pages or swap out anonymous
> > > memory. But you are right that it won't be possible in all cases, like
> > > for in kernel memory limits.
> > Such kind of memory management is less efficient than the one
> > making decisions based on global shortages and global LRU alogrithm.
>
> I also was quite surprised that openvz appears to have no way of
> constraining a container's memory usage. "I want to run this bunch of
> processes in a 4.5GB container".

I'd like to share my view on the subject of memory usage limiting.

The task of limiting a container to 4.5GB of memory bottles down to the
question: what to do when the container starts to use more than assigned
4.5GB of memory?

At this moment there are only 3 viable alternatives.

A) Have separate memory management for each container,
   with separate buddy allocator, lru lists, page replacement mechanism.
   That implies a considerable overhead, and the main challenge there
   is sharing of pages between these separate memory managers.

B) Return errors on extension of mappings, but not on page faults, where
   memory is actually consumed.
   In this case it makes sense to take into account not only the size of used
   memory, but the size of created mappings as well.
   This is approximately what "privvmpages" accounting/limiting provides in
   UBC.

C) Rely on OOM killer.
   This is a fall-back method in UBC, for the case "privvmpages" limits
   still leave the possibility to overload the system.

It would be nice, indeed, to invent something new.
The ideal mechanism would

- slow down the container over-using memory, to signal the user that
  he is over his limits,
- at the same time this slowdown shouldn't lead to the increase of memory
  usage: for example, a simple slowdown of apache web server would lead
  to the growth of the number of serving children and consumption of more
  memory while showing worse performance,
- and, at the same time, it shouldn't penalize the rest of the system from
  the performance point of view...

May be this can be achieved via carefully tuned swapout mechanism together with disk bandwidth management capable of tracking asynchronous write requests, may be something else is required.
It's really a big challenge.

Meanwhile, I guess we can only make small steps in improving Linux resource management features for this moment.

Best regards

Andrey
