Subject: Re: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Rohit Seth on Fri, 15 Sep 2006 21:58:53 GMT
View Forum Message <> Reply to Message

On Sat, 2006-09-16 at 01:21 +0400, Kir Kolyshkin wrote:
> Rohit Seth wrote:
> > On Fri, 2006-09-15 at 13:26 +0400, Kirill Korotaev wrote:
> >
> > <...skipped...>
> >
> >> for VMware which can reserve required amount of RAM for VM.
> >>
> >
> > It is much easier to provide guarantees in complete virtual
> > environments. But then you pay the cost in terms of performance.
> >
> "Complete virtual environments" vs. "contaners" is not [only] about
> performance! In the end, given a proper set of dirty and no-so-dirty
> hacks in software and hardware, their performance will be close to native.
>

I don't think there is current generation of Virtualization HW/SW that
can live with o-2% performance loss for all workloads (like the way
containers do).

> Containers vs. other virtualization types is more about utilization,
> density, scalability, portability.
>

I agree with most of it (except portability as using latest HW
technologies you can run unmodified guests in virtualized environment).

> Speaking of guarantees, yes, guarantees is easy, you just reserve such
> amount of RAM for your VM and that is all. But the fact is usually some
> part of that RAM will not be utilized by this particular VM. But since
> it is reserved, it can not be utilized by other VMs -- and we end up
> just wasting some resources. Containers, given a proper resource
> management and configuration, can have some guarantees and still be able
> to utilize all the RAM available in the system. This difference can be
> metaphorically expressed as a house divided into rooms. Dividing walls
> can either be hard or flexible. With flexible walls, room (container)
> owner have a guarantee of minimal space in your room, but if a few
> guests come for a moment, the walls can move to make more space (up to
> the limit). So the flexibility is measured as the delta between a
> guarantee and a limit.
>
> This flexibility leads to higher utilization, and this flexibility is

> one of the reasons for better density (a few times higher than that of a
> paravirtualization solution).
>

I guess as far as memory is concerned, virtualized solutions can also techniques like ballooning to oversubscribe memory. But I agree that we will almost always be able to pack things tighter in container environment.

> I will not touch scalability and portability topics here to make things
> simpler.
> > I think we should punt on hard guarantees and fractions for the first
> > draft. Keep the implementation simple.
> >
> Do I understand it right that with hard guarantees we loose the
> flexibility I have just described? If this is the case, I do not like it.

With hard guarantees, you will also end up making hooks in generic part of kernel which could be considered invasive. And yes, if you are making a hard guarantee then you will some how make sure that amount of resource is available all the time for that container. As you mentioned this is not the most optimal use of resources. And that is why I don't want to incorporate that in at least the first draft. Please look at the container kernel patches that I sent out yesterday. They allow the containers to go over board with memory as long as there is no pressure. But the moment there is any pressure on memory, pages belonging to over the limit containers get freed or swapped first.

-rohit