Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)

Posted by Rohit Seth on Fri, 08 Sep 2006 21:15:35 GMT

View Forum Message <> Reply to Message

On Fri, 2006-09-08 at 13:26 -0400, Shailabh Nagar wrote:

> Also maintenability, licensing, blah, blah.
> Replicating the software stack for each service level one
> wishes to provide, if avoidable as it seems to be, isn't such a good idea.
> Same sort of reasoning for why containers make sense compared to Xen/VMWare
> instances.
>

Having a container per service level seems like an okay thing to me.

> Memory resources, by their very nature, will be tougher to account when a
> single database/app server services multiple clients and we can essentially
> give up on that (taking the approach that only limited recharging can ever
> be achieved).

What exactly you mean by limited recharging?

As said earlier, if there is big shared segment on a server then that
can be charged to any single container. And in this case moving a task
to different container may not fetch anything useful from memory
accounting pov.

> But cpu atleast is easy to charge correctly and since that will
> also indirectly influence the requests for memory & I/O, its useful to allow
> middleware to change the accounting base for a thread/task.
>

That is not true. It depends on IO size, memory foot print etc. etc.
You can move a task to different container, but it will not be cheap.

-rohit

Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)

Posted by Shailabh Nagar on Fri, 08 Sep 2006 21:28:23 GMT

View Forum Message <> Reply to Message

Rohit Seth wrote:

>> Memory resources, by their very nature, will be tougher to account when a
>> single database/app server services multiple clients and we can essentially

>> give up on that (taking the approach that only limited recharging can ever
>> be achieved).
>
> What exactly you mean by limited recharging?
>

Memory allocated (and hence charged) by a task belonging to one container being (re)charged to another container to which task moves. Can be done but at too high a cost so not worth it most of the time.

> As said earlier, if there is big shared segment on a server then that
> can be charged to any single container. And in this case moving a task
> to different container may not fetch anything useful from memory
> accounting pov.
>
>> But cpu atleast is easy to charge correctly and since that will
>> also indirectly influence the requests for memory & I/O, its useful to allow
>> middleware to change the accounting base for a thread/task.
>>
>
> That is not true.   It depends on IO size, memory foot print etc. etc.
> You can move a task to different container, but it will not be cheap.
>
For cpu time & I/O bandwidth I disagree. Accounting to a multiplicity of
containers/BC over time shouldn't be costly.

Anyway, lets see how the implementation evolves.

> -rohit