Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Rohit Seth on Mon, 11 Sep 2006 23:58:19 GMT
View Forum Message <> Reply to Message

On Mon, 2006-09-11 at 12:42 -0700, Chandra Seetharaman wrote:
> On Mon, 2006-09-11 at 12:10 -0700, Rohit Seth wrote:
> > On Mon, 2006-09-11 at 11:25 -0700, Chandra Seetharaman wrote:

> > > There could be a default container which doesn't have any guarantee or
> > > limit.
> >
> > First, I think it is critical that we allow processes to run outside of
> > any container (unless we know for sure that the penalty of running a
> > process inside a container is very very minimal).
>
> When I meant a default container I meant a default "resource group". In
> case of container that would be the default environment. I do not see
> any additional overhead associated with it, it is only associated with
> how resource are allocated/accounted.
>

There should be some cost when you do atomic inc/dec accounting and
locks for add/remove resources from any container (including default
resource group). No?

> >
> > And anything running outside a container should be limited by default
> > Linux settings.
>
> note that the resource available to the default RG will be (total system
> resource - allocated to RGs).

I think it will be preferable to not change the existing behavior for
applications that are running outside any container (in your case
default resource group).

> >
> > > When you create containers and assign guarantees to each of them
> > > make sure that you leave some amount of resource unassigned.
> >   ^^^^^ This will force the "default" container
> > with limits (indirectly). IMO, the whole guarantee feature gets defeated
>
> You _will_ have limits for the default RG even if we don't have
> guarantees.
>
> > the moment you bring in this fuzziness.
>

> Not really.
> - Each RG will have a guarantee and limit of each resource.
> - default RG will have (system resource - sum of guarantees)
> - Every RG will be guaranteed some amount of resource to provide QoS
> - Every RG will be limited at "limit" to prevent DoS attacks.
> - Whoever doesn't care either of those set them to don't care values.
>

For the cases that put this don't care, do you depend on existing
reclaim algorithm (for memory) in kernel?

> >
> > > That
> > > unassigned resources can be used by the default container or can be used
> > > by containers that want more than their guarantee (and less than their
> > > limit). This is how CKRM/RG handles this issue.
> > >
> > >
> >
> > It seems that a single notion of limit should suffice, and that limit
> > should more be treated as something beyond which that resource
> > consumption in the container will be throttled/not_allowed.
>
> As I stated in an earlier email "Limit only" approach can prevent a
> system from DoS attacks (and also fits the container model nicely),
> whereas to provide QoS one would need guarantee.
>
> Without guarantee, a RG that the admin cares about can starve if
> all/most of the other RGs consume upto their limits.
>
> >

If the limits are set appropriately so that containers total memory
consumption does not exceed the system memory then there shouldn't be
any QoS issue (to whatever extent it is applicable for specific
scenario).

-rohit