Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Chandra Seetharaman on Mon, 11 Sep 2006 19:42:05 GMT
View Forum Message <> Reply to Message

On Mon, 2006-09-11 at 12:10 -0700, Rohit Seth wrote:
> On Mon, 2006-09-11 at 11:25 -0700, Chandra Seetharaman wrote:
> > On Fri, 2006-09-08 at 14:43 -0700, Rohit Seth wrote:
> > <snip>
> >
> > > > > > Guarantee may be one of
> > > > > >
> > > > > > 1. container will be able to touch that number of pages
> > > > > > 2. container will be able to sys_mmap() that number of pages
> > > > > > 3. container will not be killed unless it touches that number of pages
> > > > > > 4. anything else
> > > > >
> > > > > I would say (1) with slight modification
> > > > > "container will be able to touch _at least_ that number of pages"
> > > >
> > > Does this scheme support running of tasks outside of containers on the
> > > same platform where you have tasks running inside containers. If so
> > > then how will you ensure processes running out side any container will
> > > not leave less than the total guaranteed memory to different containers.
> > >
> >
> > 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.

>
> 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).
>
> > 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.

>
> > 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.

>
> -rohit
>
>
> -- -------------
> Using Tomcat but need to do more? Need to support web services, security?
> Get stuff done quickly with pre-integrated technology to make your job easier
> Download IBM WebSphere Application Server v.1.0.1 based on Apache Geronimo
> http://sel.as-us.falkag.net/sel?cmd=lnk&kid=120709&b id=263057&dat=121642
> ____
> ckrm-tech mailing list
> https://lists.sourceforge.net/lists/listinfo/ckrm-tech
--

Chandra Seetharaman | Be careful what you choose....
- sekharan@us.ibm.com | .....you may get it.
