Subject: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Chris Friesen on Wed, 01 Nov 2006 21:18:05 GMT
View Forum Message <> Reply to Message

Srivatsa Vaddagiri wrote:

>>> - Support limit (soft and/or hard depending on the resource
>>> type) in controllers. Guarantee feature could be indirectly
>>> met thr limits.

I just thought I'd weigh in on this. As far as our usage pattern is concerned, guarantees cannot be met via limits.

I want to give "x" cpu to container X, "y" cpu to container Y, and "z" cpu to container Z.

If these are percentages, x+y+z must be less than 100.

However, if Y does not use its share of the cpu, I would like the leftover cpu time to be made available to X and Z, in a ratio based on their allocated weights.

With limits, I don't see how I can get the ability for containers to make opportunistic use of cpu that becomes available.

I can see that with things like memory this could become tricky (How do you free up memory that was allocated to X when Y decides that it really wants it after all?) but for CPU I think it's a valid scenario.

Chris

Subject: Re: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by kir on Wed, 01 Nov 2006 23:01:56 GMT
View Forum Message <> Reply to Message

Chris Friesen wrote:
> Srivatsa Vaddagiri wrote:
>
>>>> - Support limit (soft and/or hard depending on the resource
>>>> type) in controllers. Guarantee feature could be indirectly
>>>> met thr limits.
>
> I just thought I'd weigh in on this. As far as our usage pattern is
> concerned, guarantees cannot be met via limits.
>
> I want to give "x" cpu to container X, "y" cpu to container Y, and "z"
> cpu to container Z.

>
> If these are percentages, x+y+z must be less than 100.
>
> However, if Y does not use its share of the cpu, I would like the
> leftover cpu time to be made available to X and Z, in a ratio based on
> their allocated weights.
>
> With limits, I don't see how I can get the ability for containers to
> make opportunistic use of cpu that becomes available.
This is basically how "cpuunits" in OpenVZ works. It is not limiting a container in any way, just assigns some relative "units" to it, with sum of all units across all containers equal to 100% CPU. Thus, if we have cpuunits 10, 20, and 30 assigned to containers X, Y, and Z, and run some CPU-intensive tasks in all the containers, X will be given 10/(10+20+30), or 20% of CPU time, Y -- 20/50, i.e. 40%, while Z gets 60%. Now, if Z is not using CPU, X will be given 33% and Y -- 66%. The scheduler used is based on a per-VE runqueues, is quite fair, and works fine and fair for, say, uneven case of 3 containers on a 4 CPU box.

OpenVZ also has a "cpulimit" resource, which is, naturally, a hard limit of CPU usage for a VE. Still, given the fact that cpunits works just fine, cpulimit is rarely needed -- makes sense only in special scenarios where you want to see how app is run on a slow box, or in case of some proprietary software licensed per CPU MHZ, or smth like that.

Looks like this is what you need, right?
> I can see that with things like memory this could become tricky (How
> do you free up memory that was allocated to X when Y decides that it
> really wants it after all?) but for CPU I think it's a valid scenario.
Yes, CPU controller is quite different of other resource controllers.

Kir.

Subject: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Paul Menage on Wed, 01 Nov 2006 23:48:21 GMT
View Forum Message <> Reply to Message

On 11/1/06, Chris Friesen <cfriesen@nortel.com> wrote:
>
> I just thought I'd weigh in on this. As far as our usage pattern is
> concerned, guarantees cannot be met via limits.
>
> I want to give "x" cpu to container X, "y" cpu to container Y, and "z"
> cpu to container Z.

I agree that these are issues - but they don't really affect the container framework directly.

The framework should be flexible enough to let controllers register any control parameters (via the filesystem?) that they need, but it shouldn't contain explicit concepts like guarantees and limits. Some controllers won't even have this concept (cpusets doesn't really, for instance, and containers don't have to be just to do with quantitative resource control).

I sent out a patch a while ago that showed how ResGroups could be turned into effectively a library on top of a generic container system - so ResGroups controllers could write to the ResGroups interface, and let the library handle setting up control parameters and parsing limits and guarantees. I expect the same thing could be done for UBC.

Paul

Subject: Re: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Matt Helsley on Thu, 02 Nov 2006 00:31:37 GMT
View Forum Message <> Reply to Message

On Thu, 2006-11-02 at 02:01 +0300, Kir Kolyshkin wrote:
> Chris Friesen wrote:
> > Srivatsa Vaddagiri wrote:
> >
> >>>> - Support limit (soft and/or hard depending on the resource
> >>>>   type) in controllers. Guarantee feature could be indirectly
> >>>>   met thr limits.
> >
> > I just thought I'd weigh in on this. As far as our usage pattern is
> > concerned, guarantees cannot be met via limits.
> >
> > I want to give "x" cpu to container X, "y" cpu to container Y, and "z"
> > cpu to container Z.
> >
> > If these are percentages, x+y+z must be less than 100.
> >
> > However, if Y does not use its share of the cpu, I would like the
> > leftover cpu time to be made available to X and Z, in a ratio based on
> > their allocated weights.
> >
> > With limits, I don't see how I can get the ability for containers to
> > make opportunistic use of cpu that becomes available.
> This is basically how "cpuunits" in OpenVZ works. It is not limiting a
> container in any way, just assigns some relative "units" to it, with sum
> of all units across all containers equal to 100% CPU. Thus, if we have

 So the user doesn't really specify percentage but values that feed into

ratios used by the underlying controller? If so then it's not terribly different from the "shares" of single level of Resource Groups.

Resource groups goes one step further and defines a denominator for child groups to use. This allows the shares to be connected vertically so that changes don't need to propagate beyond the parent and child groups.

> cpuunits 10, 20, and 30 assigned to containers X, Y, and Z, and run some
> CPU-intensive tasks in all the containers, X will be given
> 10/(10+20+30), or 20% of CPU time, Y -- 20/50, i.e. 40%, while Z gets

nit: I don't think this math is correct.

Shouldn't they all have the same denominator (60), or am I misunderstanding something?

If so then it should be:
X = 10/60 16.666...%
Y = 20/60 33.333...%
Z = 30/60 50.0%
Total: 100.0%

> 60%. Now, if Z is not using CPU, X will be given 33% and Y -- 66%. The
> scheduler used is based on a per-VE runqueues, is quite fair, and works
> fine and fair for, say, uneven case of 3 containers on a 4 CPU box.

<snip>

Cheers,
-Matt Helsley

Subject: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Chris Friesen on Thu, 02 Nov 2006 03:28:37 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:

> The framework should be flexible enough to let controllers register
> any control parameters (via the filesystem?) that they need, but it
> shouldn't contain explicit concepts like guarantees and limits.

If the framework was able to handle arbitrary control parameters, that would certainly be interesting.

Presumably there would be some way for the controllers to be called from the framework to validate those parameters?

Chris

Subject: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Paul Menage on Thu, 02 Nov 2006 07:40:43 GMT
View Forum Message <> Reply to Message

On 11/1/06, Chris Friesen <cfriesen@nortel.com> wrote:
> Paul Menage wrote:
>
> > The framework should be flexible enough to let controllers register
> > any control parameters (via the filesystem?) that they need, but it
> > shouldn't contain explicit concepts like guarantees and limits.
>
> If the framework was able to handle arbitrary control parameters, that
> would certainly be interesting.
>
> Presumably there would be some way for the controllers to be called from
> the framework to validate those parameters?

The approach that I had in mind was that each controller could register what ever control files it wanted, which would appear in the filesystem directories for each container; reads and writes on those files would invoke handlers in the controller. The framework wouldn't care about the semantics of those control files. See the containers patch that I posted last month for some examples of this.

Paul

Subject: Re: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by kir on Thu, 02 Nov 2006 08:34:42 GMT
View Forum Message <> Reply to Message

Matt Helsley wrote:
> On Thu, 2006-11-02 at 02:01 +0300, Kir Kolyshkin wrote:
>
>> cpuunits 10, 20, and 30 assigned to containers X, Y, and Z, and run some
>> CPU-intensive tasks in all the containers, X will be given
>> 10/(10+20+30), or 20% of CPU time, Y -- 20/50, i.e. 40%, while Z gets
>>
>
> nit: I don't think this math is correct.
>
> Shouldn't they all have the same denominator (60), or am I
> misunderstanding something?

>
> If so then it should be:
> X = 10/60 16.666...%
> Y = 20/60 33.333...%
> Z = 30/60 50.0%
> Total: 100.0%
>
Ughm. You are totally correct of course, I must've been very tired
yesterday night :-\
