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.