Subject: Re: [ckrm-tech] [RFC] [PATCH 0/3] Add group fairness to CFS
Posted by William Lee Irwin III on Thu, 31 May 2007 09:15:34 GMT
View Forum Message <> Reply to Message

On Thu, May 31, 2007 at 02:03:53PM +0530, Srivatsa Vaddagiri wrote:
>> Its ->wait_runtime will drop less significantly, which lets it be
>> inserted in rb-tree much to the left of those 1000 tasks (and which
>> indirectly lets it gain back its fair share during subsequent
>> schedule cycles).
>> Hmm ..is that the theory?

On Thu, May 31, 2007 at 02:26:00PM +0530, Srivatsa Vaddagiri wrote:
> My only concern is the time needed to converge to this fair
> distribution, especially in face of fluctuating workloads. For ex: a
> container who does a fork bomb can have a very adverse impact on
> other container's fair share under this scheme compared to other
> schemes which dedicate separate rb-trees for differnet containers
> (and which also support two level hierarchical scheduling inside the
> core scheduler).
> I am inclined to have the core scheduler support atleast two levels
> of hierarchy (to better isolate each container) and resort to the
> flattening trick for higher levels.

Yes, the larger number of schedulable entities and hence slower convergence to groupwise weightings is a disadvantage of the flattening. A hybrid scheme seems reasonable enough. Ideally one would chop the hierarchy in pieces so that n levels of hierarchy become k levels of n/k weight-flattened hierarchies for this sort of attack to be most effective (at least assuming similar branching factors at all levels of hierarchy and sufficient depth to the hierarchy to make it meaningful) but this is awkward to do. Peeling off the outermost container or whichever level is deemed most important in terms of accuracy of aggregate enforcement as a hierarchical scheduler is a practical compromise.

Hybrid schemes will still incur the difficulties of hierarchical scheduling, but they're by no means insurmountable. Sadly, only complete flattening yields the simplifications that make task group weighting enforcement orthogonal to load balancing and the like. The scheme I described for global nice number behavior is also not readily adaptable to hybrid schemes.

-- wli

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: [ckrm-tech] [RFC] [PATCH 0/3] Add group fairness to CFS
Posted by Srivatsa Vaddagiri on Thu, 31 May 2007 09:36:22 GMT
View Forum Message <> Reply to Message

On Thu, May 31, 2007 at 02:15:34AM -0700, William Lee Irwin III wrote:
> Yes, the larger number of schedulable entities and hence slower
> convergence to groupwise weightings is a disadvantage of the flattening.
> A hybrid scheme seems reasonable enough.

Cool! This puts me back on track to implement hierarchical scheduling in CFS :)

Once this is done and once I can get containers running on a box, I will experiment with the flattening trick for user and process levels inside containers.

Thanks for your feedback so far!

> Ideally one would chop the
> hierarchy in pieces so that n levels of hierarchy become k levels of n/k
> weight-flattened hierarchies for this sort of attack to be most effective
> (at least assuming similar branching factors at all levels of hierarchy
> and sufficient depth to the hierarchy to make it meaningful) but this is
> awkward to do. Peeling off the outermost container or whichever level is
> deemed most important in terms of accuracy of aggregate enforcement as
> a hierarchical scheduler is a practical compromise.
>
> Hybrid schemes will still incur the difficulties of hierarchical
> scheduling, but they're by no means insurmountable. Sadly, only
> complete flattening yields the simplifications that make task group
> weighting enforcement orthogonal to load balancing and the like. The
> scheme I described for global nice number behavior is also not readily
> adaptable to hybrid schemes.

--
Regards,
vatsa

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers