Subject: Re: [ckrm-tech] [RFC] [PATCH 0/3] Add group fairness to CFS
Posted by William Lee Irwin III on Wed, 30 May 2007 02:48:15 GMT
View Forum Message <> Reply to Message

William Lee Irwin III wrote:
>> Lag is the deviation of a task's allocated CPU time from the CPU time
>> it would be granted by the ideal fair scheduling algorithm (generalized
>> processor sharing; take the limit of RR with per-task timeslices
>> proportional to load weight as the scale factor approaches zero).

On Wed, May 30, 2007 at 10:09:28AM +1000, Peter Williams wrote:
> Over what time period does this operate?

Over a period of time while the task is runnable.

William Lee Irwin III wrote:
>> Negative lag reflects receipt of excess CPU time. A close-to-canonical
>> "fairness metric" is the maximum of the absolute values of the lags of
>> all the tasks on the system. The "signed minimax pseudonorm" is the
>> largest lag without taking absolute values; it's a term I devised ad
>> hoc to describe the proposed algorithm.

On Wed, May 30, 2007 at 10:09:28AM +1000, Peter Williams wrote:
> So what you're saying is that you think dynamic priority (or its
> equivalent) should be used for load balancing instead of static priority?

It doesn't do much in other schemes, but when fairness is directly
measured by the dynamic priority, it is a priori more meaningful.
This is not to say the net effect of using it is so different.

William Lee Irwin III wrote:
>> I've presented a coherent
>> algorithm. It may be that there's no demonstrable problem to solve.
>> On the other hand, if there really is a question as to how to load
>> balance in the presence of tasks pinned to cpus, I just answered it.
>
On Wed, May 30, 2007 at 10:09:28AM +1000, Peter Williams wrote:
> Unless I missed something there's nothing in your suggestion that does
> anything more about handling pinned tasks than is already done by the
> load balancer.

There are two things, both of which are relatively subtle and
coincidentally happen to some extent. The first is the unpinned lag,
which behaves much differently from unpinned load weight even if it's
not so different in concept, but apparently achieves a similar result.
The second is the relativistic point of view, which happens somewhat by

coincidence anyway, but isn't formalized anywhere at all as a basis for handling tasks pinned to cpus. The first difference is minor and the second formalizing something that mostly or completely already happens.

William Lee Irwin III wrote:
>> There was a second issue raised to which I responded. I didn't stray
>> per se. I addressed a second topic in the post.

On Wed, May 30, 2007 at 10:09:28AM +1000, Peter Williams wrote:
> OK.
> To reiterate, I don't think that my suggestion is really necessary. I
> think that the current load balancing (stand fast a small bug that's
> being investigated) will come up with a good distribution of tasks to
> CPUs within the constraints imposed by any CPU affinity settings.

It's sort of like performance. If the numbers are already good enough, my algorithm on that front need not be bothered with either.

-- wli

Subject: Re: [ckrm-tech] [RFC] [PATCH 0/3] Add group fairness to CFS
Posted by Peter Williams on Wed, 30 May 2007 04:07:47 GMT
View Forum Message <> Reply to Message

William Lee Irwin III wrote:
> On Wed, May 30, 2007 at 10:09:28AM +1000, Peter Williams wrote:
>> So what you're saying is that you think dynamic priority (or its
>> equivalent) should be used for load balancing instead of static priority?
>
> It doesn't do much in other schemes, but when fairness is directly
> measured by the dynamic priority, it is a priori more meaningful.
> This is not to say the net effect of using it is so different.

I suspect that while it's probably theoretically better it wouldn't make much difference on a real system (probably not enough to justify any extra complexity if there were any). The exception might be on systems where there were lots of CPU intensive tasks that used relatively large chunks of CPU each time they were runnable which would give the load balancer a more stable load to try and balance. It might be worth the extra effort to get it exactly right on those systems. On most normal systems this isn't the case and the load balancer is always playing

catch up to a constantly changing scenario.

Peter
--
Peter Williams pwil3058@bigpond.net.au

"Learning, n. The kind of ignorance distinguishing the studious."
-- Ambrose Bierce

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