Subject: Re: Re: [RFC] Virtualization steps
Posted by Sam Vilain on Wed, 29 Mar 2006 11:08:49 GMT
View Forum Message <> Reply to Message

On Wed, 2006-03-29 at 13:13 +0400, Kirill Korotaev wrote:
> > Well, for instance the fair CPU scheduling overhead is so tiny it may as
> > well not be there in the VServer patch. It's just a per-vserver TBF
> > that feeds back into the priority (and hence timeslice length) of the
> > process. ie, you get "CPU tokens" which deplete as processes in your
> > vserver run and you either get a boost or a penalty depending on the
> > level of the tokens in the bucket. This doesn't provide guarantees, but
> > works well for many typical workloads.
> I wonder what is the value of it if it doesn't do guarantees or QoS?

It still does "QoS". The TBF has a "fill rate", which is basically N tokens per M jiffies. Then you just set the size of the "bucket", and the prio bonus given is between -5 (when bucket is full) and +15 (when bucket is empty). The normal -10 to +10 'interactive' prio bonus is reduced to -5 to +5 to compensate.

In other words, it's like a global 'nice' across all of the processes in the vserver.

So, these characteristics do provide some level of guarantees, but not all that people expect. eg, people want to say "cap usage at 5%", but as designed the scheduler does not ever prevent runnable processes from running if the CPUs have nothing better to do, so they think the scheduler is broken. It is also possible with a fork bomb (assuming the absence of appropriate ulimits) that you start enough processes that you don't care that they are all effectively nice +19.

Herbert later made it add some of these guarantees, but I believe there is a performance impact of some kind.

> In our experiments with it we failed to observe any fairness.

Well, it does not aim to be 'fair', it aims to be useful for allocating CPU to vservers. ie, if you allocate X% of the CPU in the system to a vserver, and it uses more, then try to make it use less via priority penalties - and give others shortchanged or not using the CPU very much performance bonuses. That's all.

So, if you under- or over-book CPU allocation, it doesn't work. The idea was that monitoring it could be shipped out to userland. I just wanted something flexible enough to allow virtually any policy to be put into place without wasting too many cycles.

> > How does your fair scheduler work? Do you just keep a runqueue for each

> > vps?
> we keep num_online_cpus runqueues per VPS.

Right. I considered that approach but just couldn't be bothered
implementing it, so went with the TBF because it worked and was
lightweight.

> Fairs scheduler is some kind of SFQ like algorithm which selects VPS to
> be scheduled, than standart linux scheduler selects a process in a VPS
> runqueues to run.

Right.

> > To be honest, I've never needed to determine whether its overhead is 1%
> > or 0.01%, it would just be a meaningless benchmark anyway :-). I know
> > it's "good enough for me".
> Sure! We feel the same, but people like numbers :)

Sometimes the answer has to be "mu".

Sam.