Subject: nginx, inside openvz CT, worker_cpu_affinity
Posted by Nick Knutov on Sat, 28 Jul 2012 14:34:48 GMT
View Forum Message <> Reply to Message

Hello all,

> http://nginx.org/en/docs/ngx_core_module.html#worker_cpu_aff inity
>
> worker_processes 4;
> worker_cpu_affinity 0001 0010 0100 1000;
>
> "Binds worker processes to the sets of CPUs."

Does it make sense inside OpenVZ container?

Will it work if --cpus does not specified in CT config?
How will it work, if --cpus is specified and less then physical cores?
How will it work, if --cpus is specified and CPU has hyper-threading and
1) --cpus less then cpu cores
2) or --cpus less and odd(!) (example: --cpus: 3, physical CPU cores: 4
+ HT)

--
Best Regards,
Nick Knutov

Subject: Re: nginx, inside openvz CT, worker_cpu_affinity
Posted by Andrew Vagin on Tue, 31 Jul 2012 08:52:56 GMT
View Forum Message <> Reply to Message

Here is an answer from Vladimir Davydov, who maintains scheduler in
vzkernel.

On Tue, Jul 31, 2012 at 10:29:43AM +0400, Vladimir Davydov wrote:
> http://nginx.org/en/docs/ngx_core_module.html#worker_cpu_aff inity
> >
> > worker_processes 4;
> > worker_cpu_affinity 0001 0010 0100 1000;
> >
> > "Binds worker processes to the sets of CPUs."
> > Does it make sense inside OpenVZ container?
>
> It never works. CPU affinity masks are ignored inside containers.

Subject: Re: nginx, inside openvz CT, worker_cpu_affinity
Posted by Solar Designer on Tue, 31 Jul 2012 15:14:12 GMT
View Forum Message <> Reply to Message

On Tue, Jul 31, 2012 at 12:52:56PM +0400, Andrey Vagin wrote:
> Here is an answer from Vladimir Davydov, who maintains scheduler in
> vzkernel.
>
> On Tue, Jul 31, 2012 at 10:29:43AM +0400, Vladimir Davydov wrote:
> > http://nginx.org/en/docs/ngx_core_module.html#worker_cpu_aff inity
> > >
> > > worker_processes   4;
> > > worker_cpu_affinity 0001 0010 0100 1000;
> > >
> > > "Binds worker processes to the sets of CPUs."
> > > Does it make sense inside OpenVZ container?
> >
> > It never works. CPU affinity masks are ignored inside containers.

I don't know about nginx in particular, but CPU affinity masks certainly
work for me in OpenVZ containers, at least on RHEL5'ish kernels. I've
just re-tested on this one:

$ uname -mrs
Linux 2.6.18-308.4.1.el5.028stab100.2.owl1 x86_64

Specifically, sched_setaffinity() in custom code (custom patch to Apache's
suEXEC) and libgomp's GOMP_CPU_AFFINITY env var both work just fine.

Vladimir - does your comment apply to some other kernel versions?
RHEL6'ish maybe? It'd be good to know and be prepared... and it'd be
really unfortunate to lose this functionality when we finally move to
RHEL6'ish kernels (soonish).

Thanks,

Alexander

Subject: Re: nginx, inside openvz CT, worker_cpu_affinity
Posted by Vladimir Davydov on Tue, 31 Jul 2012 15:32:11 GMT
View Forum Message <> Reply to Message

On 07/31/2012 07:14 PM, Solar Designer wrote:
> On Tue, Jul 31, 2012 at 12:52:56PM +0400, Andrey Vagin wrote:
>> Here is an answer from Vladimir Davydov, who maintains scheduler in
>> vzkernel.
>>

>> On Tue, Jul 31, 2012 at 10:29:43AM +0400, Vladimir Davydov wrote:
>>> http://nginx.org/en/docs/ngx_core_module.html#worker_cpu_aff inity
>>> >
>>> > worker_processes 4;
>>> > worker_cpu_affinity 0001 0010 0100 1000;
>>> >
>>> > "Binds worker processes to the sets of CPUs."
>>>> Does it make sense inside OpenVZ container?
>>> It never works. CPU affinity masks are ignored inside containers.
> I don't know about nginx in particular, but CPU affinity masks certainly
> work for me in OpenVZ containers, at least on RHEL5'ish kernels. I've
> just re-tested on this one:
>
> $ uname -mrs
> Linux 2.6.18-308.4.1.el5.028stab100.2.owl1 x86_64
>
> Specifically, sched_setaffinity() in custom code (custom patch to Apache's
> suEXEC) and libgomp's GOMP_CPU_AFFINITY env var both work just fine.
>
> Vladimir - does your comment apply to some other kernel versions?
> RHEL6'ish maybe? It'd be good to know and be prepared... and it'd be
> really unfortunate to lose this functionality when we finally move to
> RHEL6'ish kernels (soonish).

In RHEL5-based kernels we had the notion of virtual cpus: tasks were
scheduled on vcpus while the vcpus were somehow distributed among
physical cpus. The sched_setaffinity syscall could be used to bind tasks
to vcpus then.

This concept was cumbersome and often sub-optimal so in RHEL6 we decided
to drop it: currently setting nr_cpus limit for a container is actually
equivalent to setting cpulimit. The decision is justified by the fact
that the latest Linux scheduler is smart enough to gather actively
interacting tasks together so that there is no need to limit parallelism
artificially.

As a result, cpu affinity support was dropped. Tasks can still use the
sched_setaffinity syscall, but it will be ignored.

>
> Thanks,
>
> Alexander

Subject: Re: nginx, inside openvz CT, worker_cpu_affinity
Posted by Vladimir Davydov on Tue, 31 Jul 2012 18:41:53 GMT

[Re-sending as it seems that for some reason the message was not delivered to the list]

On 07/31/2012 07:14 PM, Solar Designer wrote:
> On Tue, Jul 31, 2012 at 12:52:56PM +0400, Andrey Vagin wrote:
>> Here is an answer from Vladimir Davydov, who maintains scheduler in
>> vzkernel.
>>
>> On Tue, Jul 31, 2012 at 10:29:43AM +0400, Vladimir Davydov wrote:
>>> http://nginx.org/en/docs/ngx_core_module.html#worker_cpu_aff inity
>>>>
>>>> worker_processes   4;
>>>> worker_cpu_affinity 0001 0010 0100 1000;
>>>>
>>>> "Binds worker processes to the sets of CPUs."
>>>> Does it make sense inside OpenVZ container?
>>> It never works. CPU affinity masks are ignored inside containers.
> I don't know about nginx in particular, but CPU affinity masks certainly
> work for me in OpenVZ containers, at least on RHEL5'ish kernels. I've
> just re-tested on this one:
>
> $ uname -mrs
> Linux 2.6.18-308.4.1.el5.028stab100.2.owl1 x86_64
>
> Specifically, sched_setaffinity() in custom code (custom patch to Apache's
> suEXEC) and libgomp's GOMP_CPU_AFFINITY env var both work just fine.
>
> Vladimir - does your comment apply to some other kernel versions?
> RHEL6'ish maybe? It'd be good to know and be prepared... and it'd be
> really unfortunate to lose this functionality when we finally move to
> RHEL6'ish kernels (soonish).

In RHEL5-based kernels we had the notion of virtual cpus: tasks were
scheduled on vcpus while the vcpus were somehow distributed among
physical cpus. The sched_setaffinity syscall could be used to bind tasks
to vcpus then.

This concept was cumbersome and often sub-optimal so in RHEL6 we decided
to drop it: currently setting nr_cpus limit for a container is actually
equivalent to setting cpulimit. The decision is justified by the fact
that the latest Linux scheduler is smart enough to gather actively
interacting tasks together so that there is no need to limit parallelism
artificially.

As a result, cpu affinity support was dropped. Tasks can still use the
sched_setaffinity syscall, but it will be ignored.

>

> Thanks,
>
> Alexander