Subject: Re: System hangs
Posted by unilynx on Mon, 26 Feb 2007 10:27:41 GMT
View Forum Message <> Reply to Message

Kirill Korotaev wrote:
>> Interestingly, the VZs running on the machine still work, I can run
>> commands in them and they report no uptime.
>>
> sorry, what do you mean by this?
> you system hangs, but VEs still work and you can login to them? or what?
>
I tracked down parts of the problem during the message - i thought the
system hanged, but it turned out to be a PATH line which pointed to the
NFS mount: so it probably was NFS after all.
>
>> I run the vzs on reiserfs/ext3 partitions, mounted over AoE. I have the
>> feeling the kernel might actually be hanging over NFS (I use NFS to
>> share configuration and administrative files for openvz, but not for the
>> VZs themselves: running VZ on NFS mounts didn't work), but restarting
>> the NFS server doesn't help anything. I rebooted one of the hanging
>> servers, and it could access the NFS just fine afterwards, so NFS itself
>> seems to be up.
>>
> So NFS servers did hang or you just rebooted it in case?
>
The NFS server itself worked fine. I didn't reboot the NFS server: first
i restarted one of the two clients (as in: a host running only openvz
and mounting the VZ data as a client) and confirmed that the rebooted
could access the NFS server. Then, i restarted the nfs services on the
nfs server, and confirmed that the rebooted client still worked, but
that the other client (the other host running openvz, which i just left
alone while trying to debug the problem) rwas still hanging.
> AFAIK NFS clients are not always successfully survive NFS server reboot :/
> How do you mount your NFS mount? with softmounts?
>
I manually mounted the NFS filesystems after booting (using just
'mount'), and then manually started the openvz VEs.

>
>> syslog still worked, and I grabbed the following callstacks using sysrq
>> - I noticed at lot of cron processes hanging with this trace:
>>
>> Feb 23 11:31:36 web2 kernel: cron S 0000807940a0
>> 000001011ae0c050 0 3018 6456 3022 3019 3014 (NOTLB)
>> Feb 23 11:31:36 web2 kernel: 0000010119c23df8 0000000000000006
>> 000001013f674f00 ffffffffa012706b
>> Feb 23 11:31:36 web2 kernel: 0000000000000000 ffffffff8017c62b

>> ffffffff8054bc80 0000000000000000
>> Feb 23 11:31:36 web2 kernel: 000001011ae0c050 0000807940a0edd0
>> Feb 23 11:31:36 web2 kernel: Call Trace: [<ffffffffa012706b>]
>> :simfs:sim_systemcall+0x6b/0x280
>> Feb 23 11:31:36 web2 kernel: [<ffffffff8017c62b>] do_wp_page+0x44b/0x4c0
>> Feb 23 11:31:36 web2 kernel: [<ffffffff8019ceb0>] pipe_wait+0xa0/0xf0
>> Feb 23 11:31:36 web2 kernel: [<ffffffff8013b8a0>]
>> autoremove_wake_function+0x0/0x30
>>
> This calltrace looks not full.
I still have the rest of it, decided to snip it here to keep it short.

> Anyway, looks like cron is simply
> sleeping waiting on the pipe end. i.e. waiting for it's child
> to write something to the pipe.
>
Okay. I thought it might be something interesting, because the cron was
in 'simfs:sim_systemcall', and afaik that is openvz specific?

> Can you press Altsysrq-T/AltSysRq-P and provide it's full output?
>
I can, but it's about 400KB (18KB compressed, only the sysrq parts) so
I'd rather not post it to a mailinglist. But if you wish, I can put it
online or email it separately?

> Also is the kernel compiled by your self or the binary one from openvz.org?
>
 From openvz.org, the x86_64-smp one.

>> None of the VZs should be running crontab as far as I know, so this
>> should be the crontab of the underlying system. I'm not sure if it
>> should even be in a simfs function?
>>
> VEs can run crons, it's fine.
>
I don't doubt it :) With 'none of the VZs should', I meant that I had
not enabled crond on any of the VEs.

>> I think these are the crons that invoke vpsnetclean and vpsreboot (which
>> also occur a lot in the process list), so this probably explains the >80
>> load.
>>
> which load are you talking about? load average shown by top?
> load average doesn't account for processes in S state, so you cron
> doesn't influence loadavg. It accounts for only R and D state proccesses.
>
Yes, top load average.

>
>> The stack trace of vpsreboot:
>> Feb 23 11:31:44 web2 kernel: vpsreboot   D 00008ad76e6a
>> 0000010117e6e3d0   0 4316   4315   (NOTLB)
>> Feb 23 11:31:44 web2 kernel: 0000010117db9928 0000000000000006
>> 0000000000000003 ffffffff8016f624
>> Feb 23 11:31:44 web2 kernel: 000001000000f380 0000000000000202
>> ffffffff8054bc80 0000000000000000
>> Feb 23 11:31:44 web2 kernel: 0000010117e6e3d0 00008ad76e6abd1c
>> Feb 23 11:31:44 web2 kernel: Call Trace: [<ffffffff8016f624>]
>> __alloc_collect_stats+0x54/0xc0
>> Feb 23 11:31:44 web2 kernel: [<ffffffffa00b2ec1>]
>> :sunrpc:rpc_sleep_on+0x41/0x70
>> Feb 23 11:31:44 web2 kernel: [<ffffffffa00b3bd0>]
>> :sunrpc:__rpc_execute+0x1f0/0x3c0
>> Feb 23 11:31:44 web2 kernel: [<ffffffff8013b8a0>]
>> autoremove_wake_function+0x0/0x30
>> Feb 23 11:31:44 web2 kernel: [<ffffffffa00b36c7>]
>> :sunrpc:rpc_init_task+0x157/0x1f0
>> Feb 23 11:31:44 web2 kernel: [<ffffffff8013b8a0>]
>> autoremove_wake_function+0x0/0x30
>> Feb 23 11:31:44 web2 kernel: [<ffffffffa00ae8d2>]
>> :sunrpc:rpc_call_sync+0x82/0xc0
>> Feb 23 11:31:44 web2 kernel: [<ffffffffa00fa41e>]
>> :nfs:nfs3_rpc_wrapper+0x2e/0x90
>> Feb 23 11:31:44 web2 kernel: [<ffffffffa00fabe9>]
>> :nfs:nfs3_proc_access+0x109/0x180
>>
> and here is such a proccess.
> it sleeps in NFS code. So it looks like an NFS bug - client didn't restored
> after NFS server reboot.
>
The NFS server wasn't rebooted before the hang, Noone was even near it
until the openvz machines failed :)
>> Any idea what I can do to investigate this further? Could putting
>> /etc/vz and /etc/sysconfig/vz-scripts on NFS be the source of the problems ?
>>
> looks like it is :/ You can try mounting NFS with soft or intr.
> this will make sure that NFS fails with errors in case of problems instead
> of infinite hangs.
>
The clients defaulted to TCP, hard mounts. I've switched over to UDP,
soft mounts, 8KB block sizes, and enabled jumbo frames as specified by
the NFS manpages. The systems survived over the weekend. I'll switch
one back to the original TCP setup, and leave one at UDP, to verify that
its indeed the cause.