
Subject: Re: Making a VE use a particular cpu
Posted by [piavlo](#) on Tue, 23 Dec 2008 16:19:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

maratrus wrote on Tue, 23 December 2008 16:48Hello,

as far as I understand (and please correct me if I'm wrong) cpuset changes task_struct->cpu_allowed field according to its rules.

That's mean that we bind the task not to physical but virtual cpus (it is OpenVZ specific feature).

Even if I do it in HN = VE0?

If i have 8 pcpus does it means that there are also exactly 8 vcpus?

Can you elaborate (or point me to documentstion) on why this is needed in openvz kernel. Since using cpu affinity in openvz kernels is also usefull feature, and I don't understand why this vcpu & pcpu thing needed in VE0.

Quote:

Virtual CPUs are bound to physical CPUs but this connection is not permanent. So, your task could be bound to virtual CPU (vcpu) but actually run on different physical CPUs (pcpu).

You mean that the cpus i see inside HN , for example using htop are also virtual, so if i monitnor a specific process bound to specific cpu using htop, I won't notice if it migrates to other pcpu(unless if it migrates to other vcpu)? While on vanilla kernel i can see if the process migrates to other cpu.

What about mpstat command, do i see vcpus with it or pcpus?
Is there a cpu load monitoring command what will show me pcpus?

From what it looks the sys entries at
/sys/devices/system/cpu/cpu?/ show me pcpus
since putting offline cpu with:
echo 0 > /sys/devices/system/cpu/cpu3/online

Removes cpu 3 from /proc/cpuinfo , which I hope shows me pcpus!

Quote:

We can conduct the following experiment (on the test node!)
Set appropriate sets with cset command as described in previous post.
Run our VE and issue inside that VE the command e.g.
cat /dev/zero > /dev/null &

then on the HN trigger several times:

echo "p" > /proc/sysrq-trigger

and look in dmesg. There should be appear information concerning with the task running on that moment (physical and virtual CPUs numbers also should be shown). Our goal is catch "cat" process inside VE running on different physical CPUs.

Is cpu 6 below a vcpu or pcpu?

----- IPI show regs -----

CPU 6:

Modules linked in: simfs vznetdev vzethdev vzrst vzcpt tun vzdquota vzmon vzdev i2c_i801 i2c_core button

Pid: 32363, comm: cat Not tainted 2.6.24-openvz-006-r5 #1 ovz006

RIP: 0010:[<ffffff80381a06>] [<ffffff80381a06>] __clear_user+0x16/0x40

RSP: 0018:ffff81023df8bee0 EFLAGS: 00000212

RAX: 0000000000000008 RBX: 0000000000001000 RCX: 00000000000000ef

RDX: 0000000000000000 RSI: 0000000000000000 RDI: 000000000060b888

RBP: 000000000060b000 R08: 00002acf45daf070 R09: ffff8101a6c73400

R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000001000

R13: 0000000000000000 R14: 0000000000001000 R15: 0000000000000000

FS: 00002acf45db36f0(0000) GS:ffff81043fc09c00(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 000000008005003b

CR2: 0000000000690000 CR3: 0000000435a2d000 CR4: 00000000000006e0

DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000

DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400

I've made about 20 tries with the sysrq thing and the 32326 task was always running on cpu 6 according to dmesg.

So to force migration i put cpu 6 offline
echo 0 > /sys/devices/system/cpu/cpu6/online

In htop i saw that the process migrated to cpu0 (or vcpu0 more correctly).

The next time i issued sysrq the machine froze Probably sysrq is a bad thing to do with offline cpus.

AFAIU from what I've seen by default vcpu0 is bound to pcpu0
vcpu1 to pcpu1 and etc...

And I saw that vcpus default mapping was NOT altered , during my sysrq test.(can you suggest on which loads this mapping will be altered?) So cpu affinity might probably work as should in 99% of the time in openvz