

Kirill,

> I would be thankful if you find/provide some exact way
> to reproduce it. If not - we will play a bit later and check it.
I just tried it like described below, and got result in ~1hour, I'll
try tomorrow to find faster/more exact way of causing this, but right
now what causes fairly consistently such result is this:

- I've got debian guest, with dev packages (kernel-package), and kernel
sources sitting ready to compile:
- ssh to guest, export CONCURRENCY_LEVEL=6; fakeroot make-kpkg
- from HN, reduce CPUs to less then CONCURRENCY_LEVEL, for example to 1

when the make-kpkg finishes, what I have is guest that reports load_avg
~6:

```
eyck@etchdev386:~/40p-ovz/work$ w
15:39:52 up 4:47, 1 user, load average: 6.00, 6.01, 5.91
and host which correctly reports load ~0:
codev64:~# w
15:44:18 up 4:55, 2 users, load average: 0.00, 0.00, 0.24
```

I guess that what causes it, is having in guest more processes with runnable
state then reduced virtual cpus available.

> The issue is not major, it's most likely screw up of process
> counters which doesn't affect anything but statistics.
yupp, it's almost cosmetic, but still, it's nicer to know that you're
running code with little amount of warts, and also for people who monitor
their guests from inside - this might result in unnecessary nightly alerts.

> Kirill
> P.S. philosophical questions was due to the fact that
> top and similar tools don't show processes which took little
> CPU time and die, thus you can have 100% CPU load and
> top still doesn't show any processes consuming CPU.
but then you should have this load visible on HN, and tools like sa should
catch it (although I ain't got sa statistics on machine I detected this
available).

--

Key fingerprint = 40D0 9FFB 9939 7320 8294 05E0 BCC7 02C4 75CC 50D9
Total Existence Failure
