

## Software Interrupts and SMP affinity

Although the IPv6-only network is arranged to promote native v6 routing over everything, the fact of the matter is that poorly-designed software, and the habits of users will bring a certain amount of traffic to the 464XLAT/NAT64 chain, or simply to the NAT64 individually. If the translating machinery does not perform on par with native routing, that's probably acceptable, but it should perform well-enough to remain a viable option to NAT-PT and CGN.

As an informal examination of translation performance, I ran the `top` program on the NAT64 host and on the 464XLAT-CLAT host in a terminal window, and then started an SFTP transfer from a remote host to a v6-only client host using a v4 numeric address, so that the transfer was forced to use 464XLAT/NAT64.

A single transfer from the remote host to the client caused NAT64 software interrupts to occupy the single CPU core at about 9% to 12%, just from watching the display and the CLAT host showed about 7% to 10% CPU for software interrupts.

Adding a second stream, from a different remote machine to a different client pushed NAT64 to nominally 100%, while CLAT was at about 30%. Adding a third stream kept NAT64 at about 100%, while CLAT rose to about 90%.

This apparent greater-than-linear increase in CPU load merits some attention. Of course, it might be mitigated by segmentation and other offloading in the Ethernet driver, but at least segmentation offloads have been turned off per the Jool instructions.

In order to do better I shut down the NAT64 and CLAT VMs, and added 2 CPU cores per VM to give each a total of 3, and then after rebooting, I did:

```
cat /proc/interrupts
```

to find the interrupts for Ethernet adapters (ens18 on CLAT and ens18 + ens19 on NAT64),

and

```
echo 7 > /proc/irq/11/smp_affinity
```

(where "11" is the number of the interrupt you want to assign affinity to, and "7" is all three bits representing CPU cores)

to spread software interrupts across all cores.

More work is required on this.

Figure 3 shows CPU, memory and network usage during 2 runs of the 3-stream test, the first with higher CPU usage, mostly by software interrupts, and the second with lower usage after adding cores and re-configuring interrupt/CPU affinity.

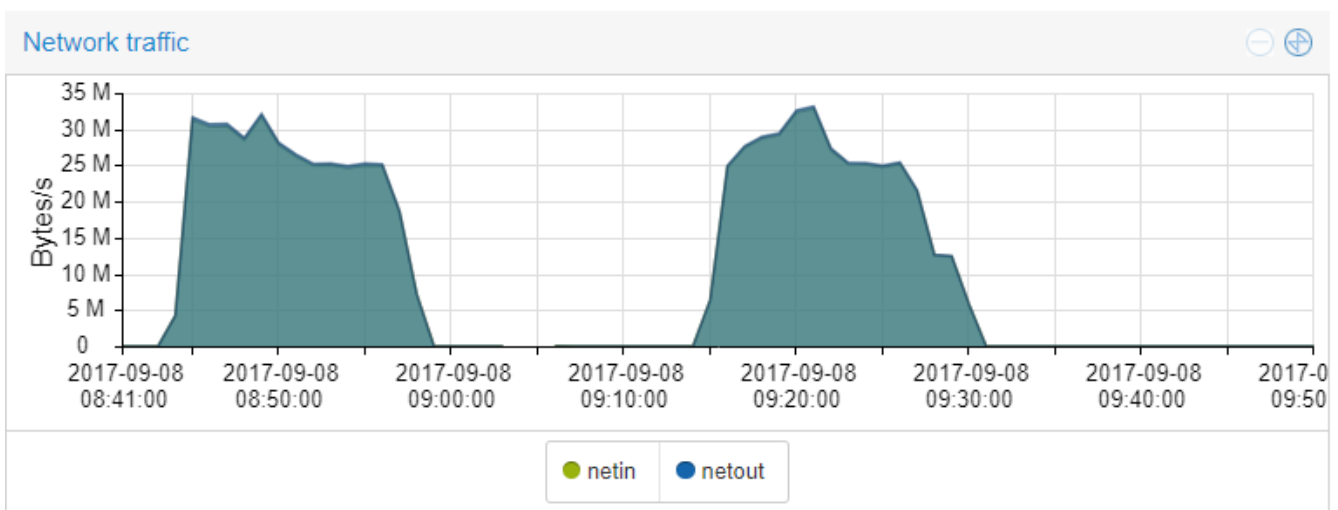
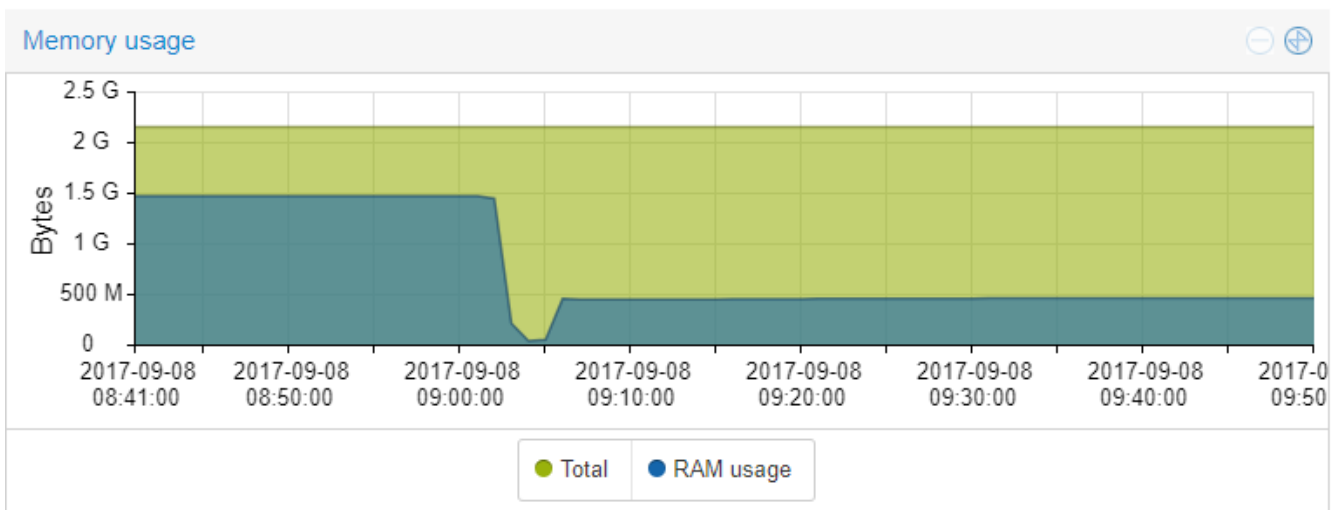
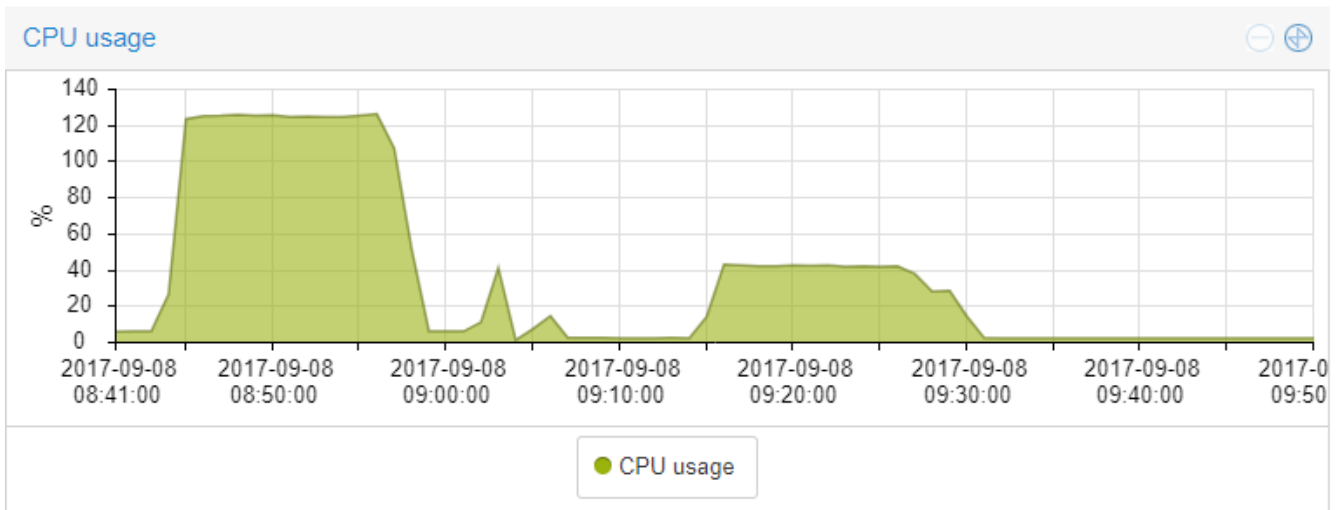


Figure 3: NAT64 CPU load with 3 ~90 Mbits/s streams, first with one core, and then with 3 cores and shared affinity among cores