Improving network performance within the cloud: eduPERT and a case study.

Alessandra Scicchitano
SWITCH
Life is full of existential questions:

- Who am I?
- Who do I work for?
- Why am I here?
"Lit Fibre" links provide multiple wavelengths currently at 10 Gbps.
I work for

The Swiss Research and Educational network

&

eduPERT

A community of Performance Experts
eduPERT is an open community to all interested in performance issues.

- An extended field of action moving towards the end user and the end point, including positioning for new technologies such as cloud which increase the complexity of performance understanding and diagnosis.

- Increased dissemination of performance expertise among NRENs and end user communities.
How we break boundaries

- Monthly phone calls
- F2F meetings
- PERT KB: http://kb.pert.geant.net/
- eduPERT Portal: http://edupert.gent.net/
- Mailing list: pert-discuss@geant.net (registration link on the portal)
- Performance U!
Performance U! stands for Performance University. Which is an expert school that aims to train performance experts, providing validated performance resources and face to face training.

“Performance U!” hosts an annual school where the eduPERT community can learn about performance implications of new tools and new technologies and a summer workshop where the community brings outside its experience and knowledge.

Because great performance doesn’t know borders or boundaries
SWITCH
● Happily based where Heidi lives 😊

● A foundation born 26 years ago

● It is the Swiss NREN but it also offers many other services including Cloud
- Project on going since September 2012

- The main goal of the project is learning as much as possible about the Cloud and its possible use within our community

- A pilot infrastructure has been built from scratch and it is now open to the community
BCC - Hardware
BCC - Software

- OpenStack
- Puppet
- Ceph-FS
- KVM
The story

• A customer from USI (Universita’ della Svizzera Italiana) running his own application within BCC noticed that the max speed he could reach was only 80Mb/s

• Doing some research on his own to understand what the problem could be he read a blog [1] about openstack and high speed network.

• In particular he asked us whether the words “virtio” and “vhost-net” would ring a bell

• It did indeed…
Going back to…

- Almost one year ago: The SWITCH DNS case
- The DNS server is on a VM
- Symptom: Terrible qps (queries per second that the server is able to answer without “dropping” )
### Before...

**Results BIND 9.8.1-P1**

<table>
<thead>
<tr>
<th></th>
<th>bamus (1 vCPU)</th>
<th></th>
<th>bamus (2 vCPU)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manaro</td>
<td>~21'000 qps</td>
<td>~16'000 qps</td>
<td>~10'000 qps</td>
<td>~18'000 qps</td>
<td>~15'000 qps</td>
</tr>
<tr>
<td>asama</td>
<td>~18'500 qps</td>
<td>~15'500 qps</td>
<td>~10'000 qps</td>
<td>~16'000 qps</td>
<td>~14'000 qps</td>
</tr>
<tr>
<td>manaro + asama</td>
<td>~21'000 qps</td>
<td>~16'000 qps</td>
<td>~10'000 qps</td>
<td>~18'000 qps</td>
<td>~15'000 qps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+querylog</td>
<td>+querylog, +dsc</td>
<td>-</td>
<td>+querylog</td>
<td>+querylog, +dsc</td>
</tr>
<tr>
<td></td>
<td>~44'000 qps</td>
<td>~38'000 qps</td>
<td>~28'000 qps</td>
<td>~32'000 qps</td>
<td>~31'000 qps</td>
</tr>
<tr>
<td></td>
<td>~42'000 qps</td>
<td>~31'000 qps</td>
<td>~35'000 qps</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Of course we were immediately able to say where the problem was:

“The bottleneck is located somewhere between the Hardware Interface of the mother host and the BIND (DNS software) in the Virtual machine”
That's a joke, right?
We tried different things before digging into the KVM architecture, but nothing really worked
Virtualization

Three different types of virtualization exist:

- Emulation
- Paravirtualization
- Hardware pass-through
KVM - Emulation
KVM - Paravirtualization
Hardware pass-through means that the guest VM has direct access to the physical hardware.
Paravirtualization was the solution to the DNS problem.
<table>
<thead>
<tr>
<th></th>
<th>simbo (1 vCPU)</th>
<th>simbo (2 vCPU)</th>
<th>simbo (3 vCPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>manaro</strong></td>
<td>23'900 qps</td>
<td>16'200 qps</td>
<td>45'800 qps</td>
</tr>
<tr>
<td></td>
<td>+querylog</td>
<td>+querylog, +dsc</td>
<td>+querylog</td>
</tr>
<tr>
<td></td>
<td>10'700 qps</td>
<td>28'500 qps</td>
<td>19'500 qps</td>
</tr>
<tr>
<td><strong>manaro + lopevi</strong></td>
<td>27'000 qps</td>
<td>17'700 qps</td>
<td>57'100 qps</td>
</tr>
<tr>
<td></td>
<td>+querylog</td>
<td>+querylog, +dsc</td>
<td>+querylog</td>
</tr>
<tr>
<td></td>
<td>11'600 qps</td>
<td>29'500 qps</td>
<td>36'400 qps</td>
</tr>
<tr>
<td></td>
<td><strong>48'800 qps</strong></td>
<td>19'500 qps</td>
<td>28'300 qps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>73'000 qps</td>
<td>39'000 qps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>30'000 qps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>50'300 qps</strong></td>
</tr>
</tbody>
</table>
We looked for the type of interface that our VMs had:
Let’s paravirtualize BCC

The max speed of the rtl8139 network adapter is 100Mb/s (?????)

No wonder why we couldn’t get any better than that.

PARAVIRTUALIZATION!
This line needs to be added in the nova.config file

```
libvirt_use_virtio_for_bridges=true
```

With this modification every VM is now created directly with paravirtualized drivers.
Howto: VMs already existing

Manual steps on the mother-host:
• adding <model type='virtio' /> to the interface in section /var/lib/nova/instances/instance-xxxx/libvirt.xml enables the virtio driver
• virsh destroy instance-xxxxxx
• virsh undefine instance-xxxxxx
• virsh define /var/lib/nova/instance-xxxxxxx/libvirt.xml
• vish start instance-xxxxxx
Drivers now

```
  sk0,format=qcow2,cache=none -device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x4,drive=drive-virtio-disk0,id=virtio-disk0,bootindex=1 -drive file=/var/lib/nova/instances/instance-00000a7/disk.local,if=none,id=drive-virtio-disk1,format=qcow2,cache=none -device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x5,drive=drive-virtio-disk1,id=virtio-disk1 -netdev tap,fd=22,id=hostnet0,vhost=on,vhostfd=23 -device virtio-net-pci netdev=hostnet0,id=net0,mac=fa:16:3e:00:00:00
```
Comparison

- RTL8139

```
root@questnet:~# iperf -i 1 -c 10.0.0.20

-----------------------------------------------
Client connecting to 10.0.0.20, TCP port 5001
TCP window size: 22.9 KByte (default)

[ 3] local 10.0.0.29 port 57754 connected with 10.0.0.20 port 5001
[ ID] Interval          Transfer     Bandwidth
[ 3]  0.0 -  1.0 sec   14.2 MBytes  120 Mbits/sec
[ 3]  1.0 -  2.0 sec   16.4 MBytes  137 Mbits/sec
[ 3]  2.0 -  3.0 sec   16.9 MBytes  142 Mbits/sec
[ 3]  3.0 -  4.0 sec   15.4 MBytes  129 Mbits/sec
[ 3]  4.0 -  5.0 sec   14.1 MBytes  118 Mbits/sec
[ 3]  5.0 -  6.0 sec   15.9 MBytes  133 Mbits/sec
[ 3]  6.0 -  7.0 sec   15.5 MBytes  130 Mbits/sec
[ 3]  7.0 -  8.0 sec   16.4 MBytes  137 Mbits/sec
[ 3] 10.0-10.0 sec    158 MBytes  132 Mbits/sec

root@questnet:~# owning 10 0 0 4
```
Comparison

- First improvement using e1000

```plaintext
ufuntu@questnet:~$ iperf -c 10.0.0.4 -i 1

Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 22.9 KByte (default)

<table>
<thead>
<tr>
<th>ID</th>
<th>Interval</th>
<th>Transfer</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.0-1.0 sec</td>
<td>11.6 MBytes</td>
<td>97.5 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>1.0-2.0 sec</td>
<td>16.1 MBytes</td>
<td>135 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>2.0-3.0 sec</td>
<td>28.9 MBytes</td>
<td>242 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>3.0-4.0 sec</td>
<td>46.0 MBytes</td>
<td>386 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>4.0-5.0 sec</td>
<td>40.5 MBytes</td>
<td>340 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>5.0-6.0 sec</td>
<td>25.5 MBytes</td>
<td>214 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>6.0-7.0 sec</td>
<td>29.9 MBytes</td>
<td>251 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>7.0-8.0 sec</td>
<td>30.4 MBytes</td>
<td>255 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>8.0-9.0 sec</td>
<td>29.6 MBytes</td>
<td>249 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>9.0-10.0 sec</td>
<td>37.5 MBytes</td>
<td>315 Mbits/sec</td>
</tr>
<tr>
<td>3</td>
<td>0.0-10.0 sec</td>
<td>296 MBytes</td>
<td>248 Mbits/sec</td>
</tr>
</tbody>
</table>
```
Comparison

- VIRTIO

```
ubuntu@questnet:~$ iperf -c 10.0.0.4 -i 1

---------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 22.9 KByte (default)
---------------------------------
[  3] local 10.0.0.29 port 48661 connected with 10.0.0.4 port 5001
 [ ID] Interval     Transfer  Bandwidth
[  3]  0.0-  1.0 sec  849 MBytes  7.12 Gbits/sec
[  3]  1.0-  2.0 sec  937 MBytes  7.86 Gbits/sec
[  3]  2.0-  3.0 sec  876 MBytes  7.34 Gbits/sec
[  3]  3.0-  4.0 sec  857 MBytes  7.19 Gbits/sec
[  3]  4.0-  5.0 sec  935 MBytes  7.84 Gbits/sec
[  3]  5.0-  6.0 sec  956 MBytes  8.02 Gbits/sec
[  3]  7.0-  8.0 sec  976 MBytes  8.18 Gbits/sec
[  3]  8.0-  9.0 sec  956 MBytes  8.02 Gbits/sec
[  3]  9.0-10.0 sec  991 MBytes  8.32 Gbits/sec
```

OpenStack Components
References
