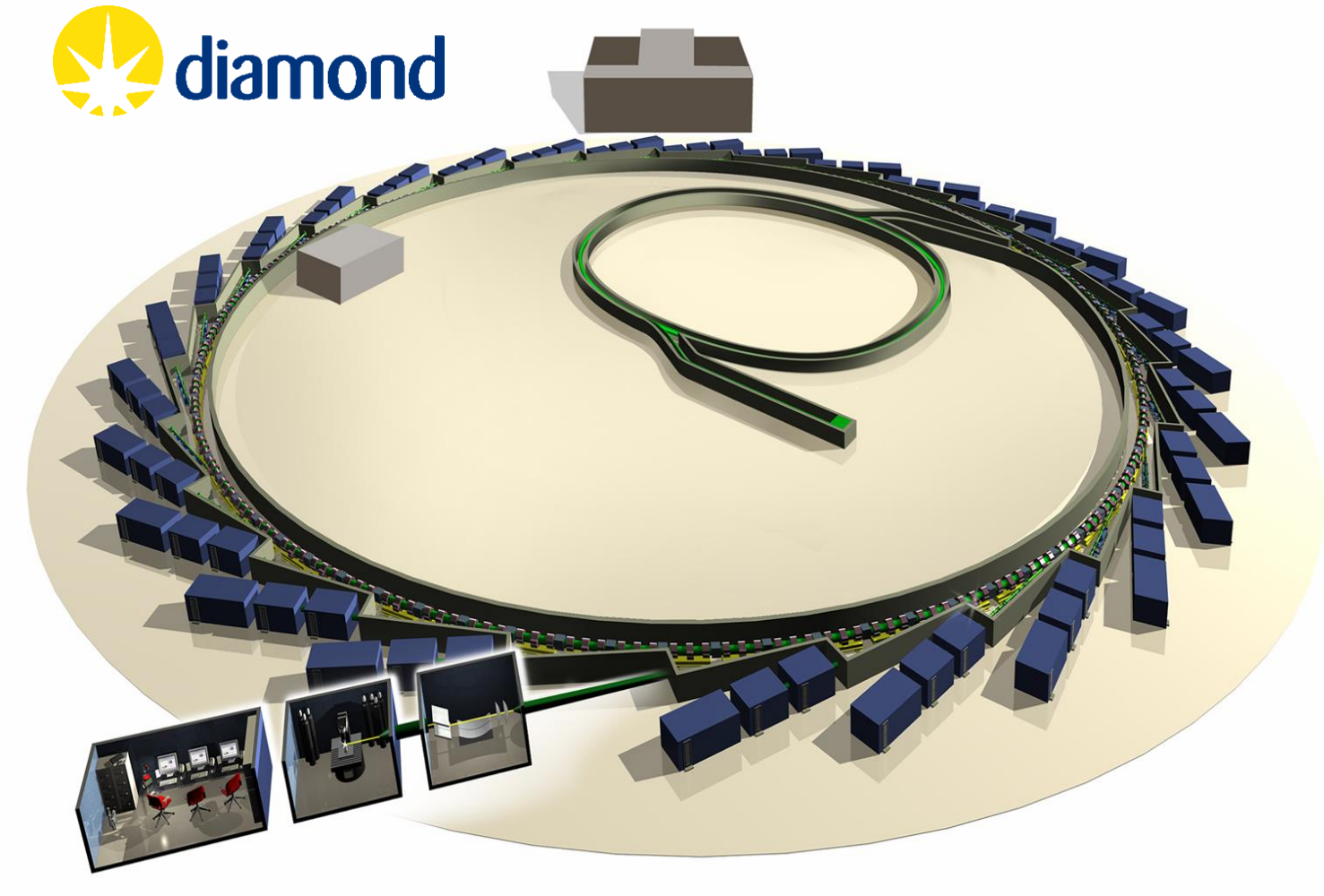
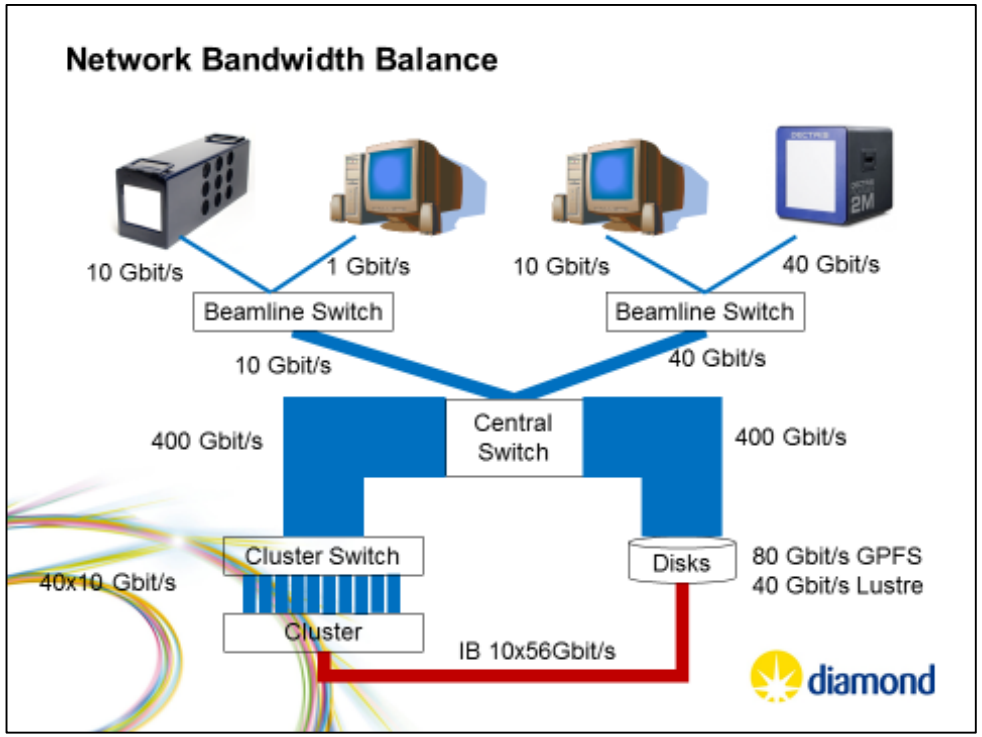
It's the network again (even on Windows)…

Spectrum Scale User Group @ SC19 Denver

Michael Hennecke | 17-Nov-2019

Motivation: Node Expels at Diamond Light Source





Two Extreme BDX8 Science Core Routers,
Avaya/Extreme Beamline Switches (mostly 10GbE)

What is **nsdperf** ?

- An IBM „as-is“ tool (in `/usr/lpp/mmfs/samples/net/`) to perform **many-to-many** network throughput testing, simulating Spectrum Scale NSD client / server I/O
  - Supports TCP sockets (all platforms), and RDMA verbs (Linux only)
  - Does not require Spectrum Scale to be installed/configured (and ignores its tunables)

- Client / server architecture:
  - Run „`nsdperf -s`“ server process on all nodes to be tested (NSD servers and NSD clients)
  - Run one additional „`nsdperf`“ client, which accepts commands to define and run the tests

- Building **`nsdperf`**:

```
cd /usr/lpp/mmfs/samples/net ; ls # makefile nsdperf.C README
g++ -O2         -o nsdperf -lpthread -lrt                    nsdperf.C # no RDMA
g++ -O2 -DRDMA -o nsdperf -lpthread -lrt -libverbs -lrdmacm nsdperf.C # RDMA
./nsdperf.exe -h # shows the cmd-line options, but not the nsdperf commands
more README      # this is the one and only nsdperf documentation
```

Running **nsdperf** (with TCP sockets) – Basic Test

- Start the server processes:

```
xdsh dss01-dss04,cli01 '/.../nsdperf -s </dev/null >/dev/null 2>&1 &'
```

- Run the test job:

```
cat <<E_O_F | /usr/lpp/mmfs/samples/net/nsdperf
server   dss01 dss02 dss03 dss04
client   cli01
ttime    60
test     nwrite
test     read
quit
E_O_F
```

- Stop the server processes:

```
xdsh dss01-dss04,cli01 'echo "kill `hostname -s`" | /.../nsdperf'
```

Running **nsdperf** – Revised Test

- Start the server processes:

```
xdsh dss01-dss04,cli01 '/.../nsdperf -s </dev/null >/dev/null 2>&1 &'
```

- Run the test job:

```
cat <<E_O_F | /usr/lpp/mmfs/samples/net/nsdperf
server   dss01 dss02 dss03 dss04
client   cli01
buffsize 16777216  #  make sure the buffer size matches the FS blocksize !
ttime    60
test     nwrite
test     read
quit
E_O_F
```

- Stop the server processes:

```
xdsh dss01-dss04,cli01 'echo "kill `hostname -s`" | /.../nsdperf'
```

Running **nsdperf** – Further Revised Test

- Start the server processes:

```
xdsh dss01-dss04,cli01 '/.../nsdperf -s </dev/null >/dev/null 2>&1 &'
```

- Run the test job:

```
cat <<E_O_F | /usr/lpp/mmfs/samples/net/nsdperf
server   dss01 dss02 dss03 dss04
client   cli01
buffsize 16777216  #  make sure the buffer size matches the FS blocksize !
threads  24        #  number of client threads should usually match #cores
ttime    60
test     nwrite
test     read
quit
E_O_F
```

- Stop the server processes:

```
xdsh dss01-dss04,cli01 'echo "kill `hostname -s`" | /.../nsdperf'
```

Zooming In ... Good and Bad Nodes

Node i13-hamamatsu01 Read [MB/s]:

BS \ threads	1M	4M	16M
1 thread	208 - 297	322 - 433	351 - 646
6 thread	320 - 673	370 - 437	320 - 431
12 thread	426 - 601	410 - 460	354 - 420
24 thread	389 - 557	281 - 510	341 - 486

Node i13-pcodimax02 Read [MB/s]:

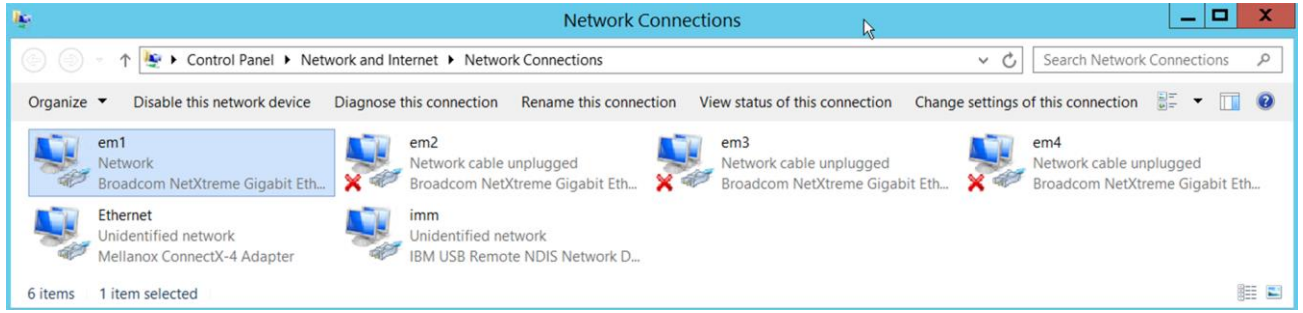
BS \ threads	256k	1M	4M	16M
1 th	180 - 410	554 - 686	**140** - 755	**1.0 - 139**
6 th	**90** - 877	**57** - 1040	**1.5 - 4.4**	**0 - 33.5**
16 th				**0** - 1110

Result ranges were obtained from running single-client nsdperf read test to each DSS-G server sequentially

The Culprit: Windows Server TCP Settings

- **Update** Windows Server's NIC **firmware and drivers** to latest level, and reboot
- Review and tune the Advanced Settings of the 10GbE Network Interface Card:

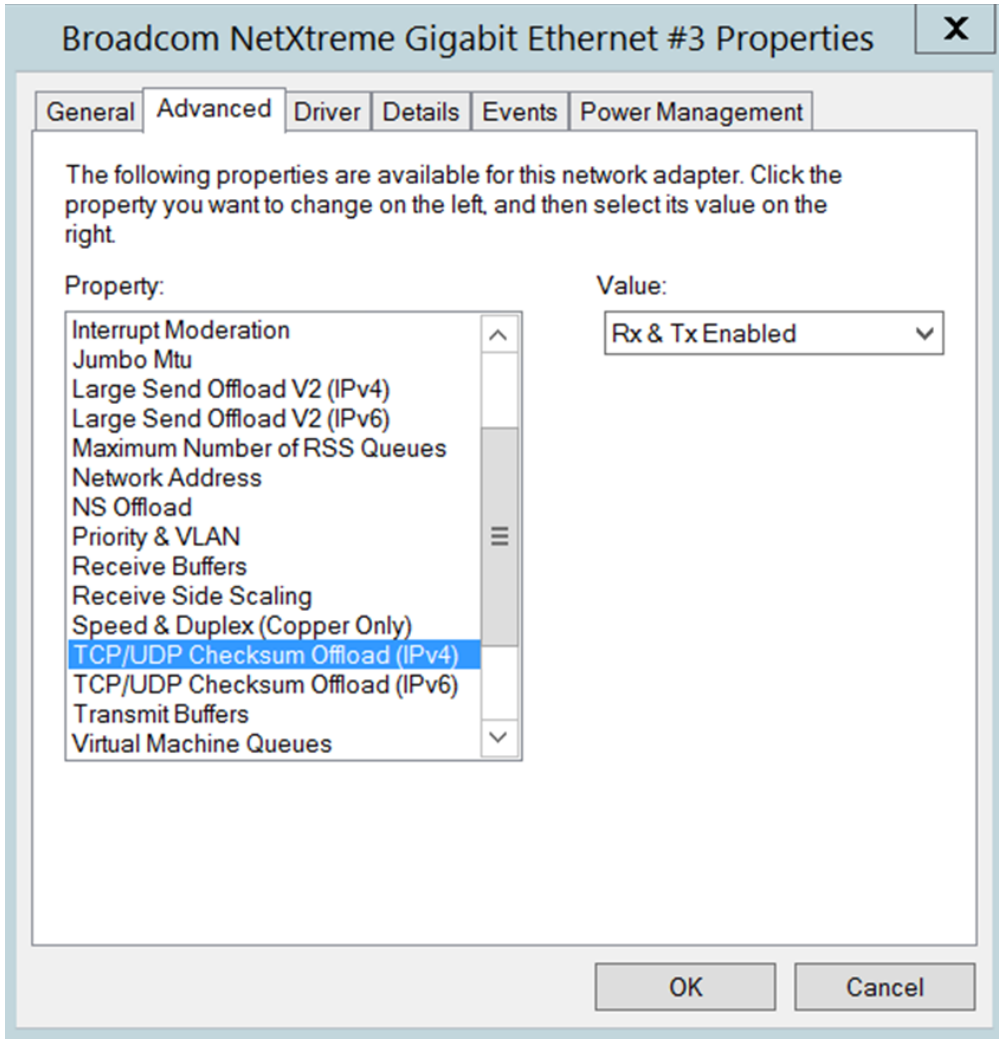
  Control Panel → Network and Internet → Network and Sharing Center → Change Adapter Settings → Right-Click on the adapter → Properties → Networking tab → Configure → Advanced tab



  – **Enable** TCP Checksum Offload (TCO)

  – **Enable** Receive Side Scaling (RSS)

➔ Sightly slower writes, but **stabilized the reads**



mhennecke @ lenovo.com

thanks.