NETWORK DIAGNOSTIC FOR STORNEXT DLC

Alain Renaud

Sustaining engineering

February 2013

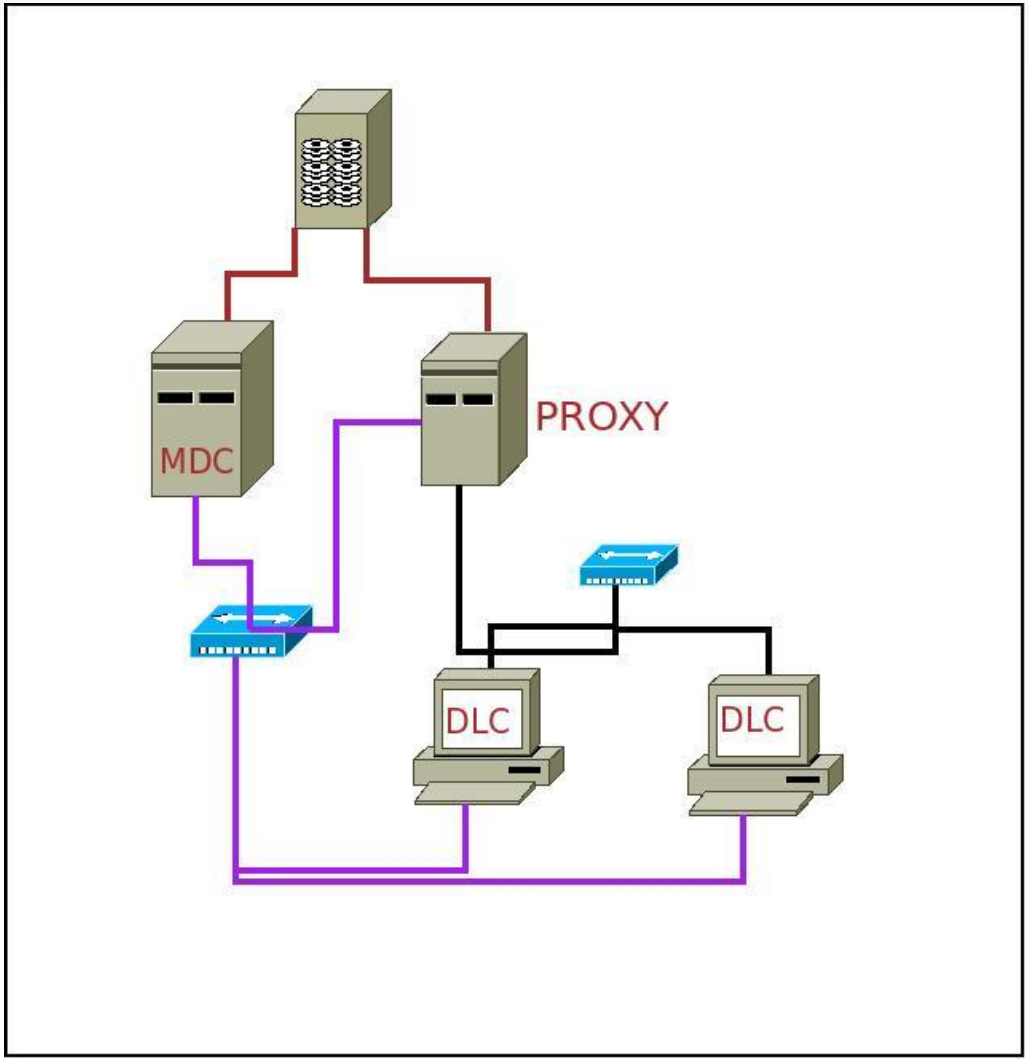
Network Diagnostic for StorNext DLC.

- Network Description.
- Understanding the dpserver file.
- Network Diagnostic tools.
  - netperf.
  - latency-test
  - Other network tools
  - iperf
- References.

Network Description.

- Need to understand the customer network configuration to be able to understand where the network bottle neck can be located.
- Tools to use for network description.
  - `netstat -nr`
  - `ifconfig -a`
  - `ipconfig /all`
  - `/usr/cvfs/config/dpserver`
- Typical config for a SNFS gateway
  - 1 network for the metadata (private)
  - 1 network for the DLC data (private)
  - 1 network for internet access (public)
  - NOTE: sometime the Metadata or DLC network is not private.

Network Description



Understanding the dpserver file.

- The SNFS nodes that are going to act as DLC servers need to have a configuration file called /usr/cvfs/config/dpserver.
- If this dpserver is not present and the fstab contain diskproxy=server then the filesystem will not mount and you will see the following error message in /usr/cvfs/debug/mount.<FS>.out

```
No Disk Proxy Server config file found.
See the sndpscfg(1) and dpserver(4) man pages for instructions on creating
one.
```

- You can use the command sndpscfg -e to create the file.
- The file contain 2 sections:
  - Tuning section: Where you can change the different tuning values.
  - Interface section: Required to specify on which interface the DLC traffic will go.

Understanding the dpserver file. (cont)

- We will now look at all the different tuning and explain what they are used for.
  - `tcp_window_size_kb`:
    - Default 64, Minimum 8, Maximum 2048
    - specifies the size in Kilobytes of the TCP window used for Proxy Client I/O connections.
  - `transfer_buffer_size_kb:`
    - Default 256, Minimum 32, Maximum 1024
    - specifies the size in Kilo-bytes of the socket transfer buffers used for Proxy Client I/O.
  - `transfer_buffer_count:`
    - Default 16, Minimum 4, Maximum 128
    - specifies the number of socket transfer buffers used per connection for Proxy Client I/O.
    - Only valid on Windows clients.

Understanding the dpserver file. (cont)

- `server_buffer_count:`
  - Default 8, Minimum 4, Maximum 32.
  - The number of I/O buffers allocated for each network interface on the gateway server. This parameter is used only by Linux servers.
- `daemon_threads:`
  - Default 8, Minimum 2, Maximum 32.
  - The maximum number of daemon threads used by the gateway server.

- On High Speed network it is recommended to use the maximum value for all parameter if possible.

Network Diagnostic tools.

- There are multiple different kind of tools to analyze network performance: nttcp, netperf, iperf.
- All these tools a very good for network diagnostic. You can select the tool depending on your preference and/or the customer requirement.
- For this presentation we will talk about netperf.

netperf

- Netperf is a very complex network diagnostic tool and it has multiple different options. Here a some point that need to be looked at.
  - Netperf has 2 binaries ‘netserver’ server code. ‘netperf’ client code.
  - Need to make sure the server and client version matches. Version 2.4 is not compatible with 2.6
  - The default test is TCP_STREAM which is sending tcp stream to the server you can reverse the direction by changing the name to TCP_MAERTS.
  - You also want to play with the different windows size to match the `tcp_window_size_kb` that you plan to use. The default window size is 64K you can use the flag ‘–S 1M –s 1M’ to set the window size to 1Meg on the server(netserver) and on the client(netperf).

netperf (cont)

- You also want to set the –D flag to specify that TCP_NODELAY is used.

- Exemples:
  - Sending data with 1Meg windows. (client -> server)

```
# netperf -H proxy-srv -p 5001 -t TCP_STREAM -- -D -S 1M -s 1M
TCP STREAM TEST from 0.0.0.0 () port 0 AF_INET to proxy-srv () port 0 AF_INET :
nodelay
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

262142 262142 262142    10.00     939.15
```

netperf (cont)

- Receiving data for 30 sec with 2Meg windows. (client <- client)

```
# netperf -l30 -H proxy-srv -p 5001 -t TCP_MAERTS -- -D -S 2M -s 2M
TCP MAERTS TEST from 0.0.0.0 () port 0 AF_INET to proxy-srv () port 0 AF_INET :
nodelay
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

262142 262142 262142    30.00     880.81
```

latency-test

- With netperf 2.6 you can also take a look at the network latency. This is more important on the Meta data network because we need answer to the message we send really fast. Some network can have a decent network speed, specially with big tcp windows but have a really bad latency.
  - Here is a quick example of a latency test.

```
$ netperf -H proxy-srv -p 5001 -j  -t omni -- -d maerts -k "MEAN_LATENCY"
OMNI Receive TEST from 0.0.0.0 () port 0 AF_INET to proxy-srv () port 0 AF_INET
MEAN_LATENCY=229.25
```

latency-test (cont)

- Stornext does provide a latency-test in ‘cvadmin’ this test tell the FSM of the filesystem to send a message to its client and calculate the response time.

```
# cvadmin -F vsop02a -e 'latency-test all'
Select FSM "vsop02a"


Test started on client 1 (vsop-rhel62-mdc.mdh.quantum.com)... latency 126us
Test started on client 4 (vsop-centos63-gw.mdh.quantum.com)... latency 375us
Test started on client 6 (vsop-centos63-clnt.mdh.quantum.com)... latency 311us
Test started on client 7 (vsop-centos63-clnt2.mdh.quantum.com)... latency 282us
```

Other network tools

- There are other tools that you can use to see how the network behave when running netperf.
  - netstat –s
    - report network statistic grab data before and after the test. For example if the number of ‘segments retransmited‘ increase drastically during the test this indicate a high packet lost usually due to defective network equipement.
  - sar –n DEV 1 30
    - this will grab the network statistic every second 30 times.
  - tcpdump -i bond0 -s 96 -w data.dump host 10.65.178.241
    - It can be very useful to grab the network data during testing so we can analyze it later.
  - Also starting a graphics network statistic tool like the ‘KDE System Monitor’ or ‘Windows task manager’ is a really good visual aid to see network problem.

iperf

- Iperf is not part of this presentation but here is a quick reference guide on how to use it.
  - iperf –s   Start a server
  - iperf –c crest -r –w 1M   Start client read/write test.
- To calculate the network latency (jitter) you can use this test.
  - iperf –s –u –i 2   Start server in UDP udapte 2 seconds.
  - iperf –c crest –u –b   Start client.
  - You then look for the jitter value on the server side.

References

- http://www.netperf.org/netperf/
- http://code.google.com/p/netperf-win/
- http://sourceforge.net/projects/iperf/

BE CERTAIN

© 2012 Quantum Corporation. Company Confidential. Forward-looking information is based upon multiple assumptions and uncertainties, does not necessarily represent the company's outlook and is for planning purposes only.