Getting the most out of Lustre with the TCP LND

Blake Caldwell (ORNL)
LUG '13 San Diego
April 17, 2013

Overview

- SockInd advantages
- Case study (SNS Lustre filesystem)
- Network verification
- NIC tuning
- TCP host kernel parameters
- LND and LNET parameters
- Lustre parameters
- LNET selftest
- Results with 2x10GE

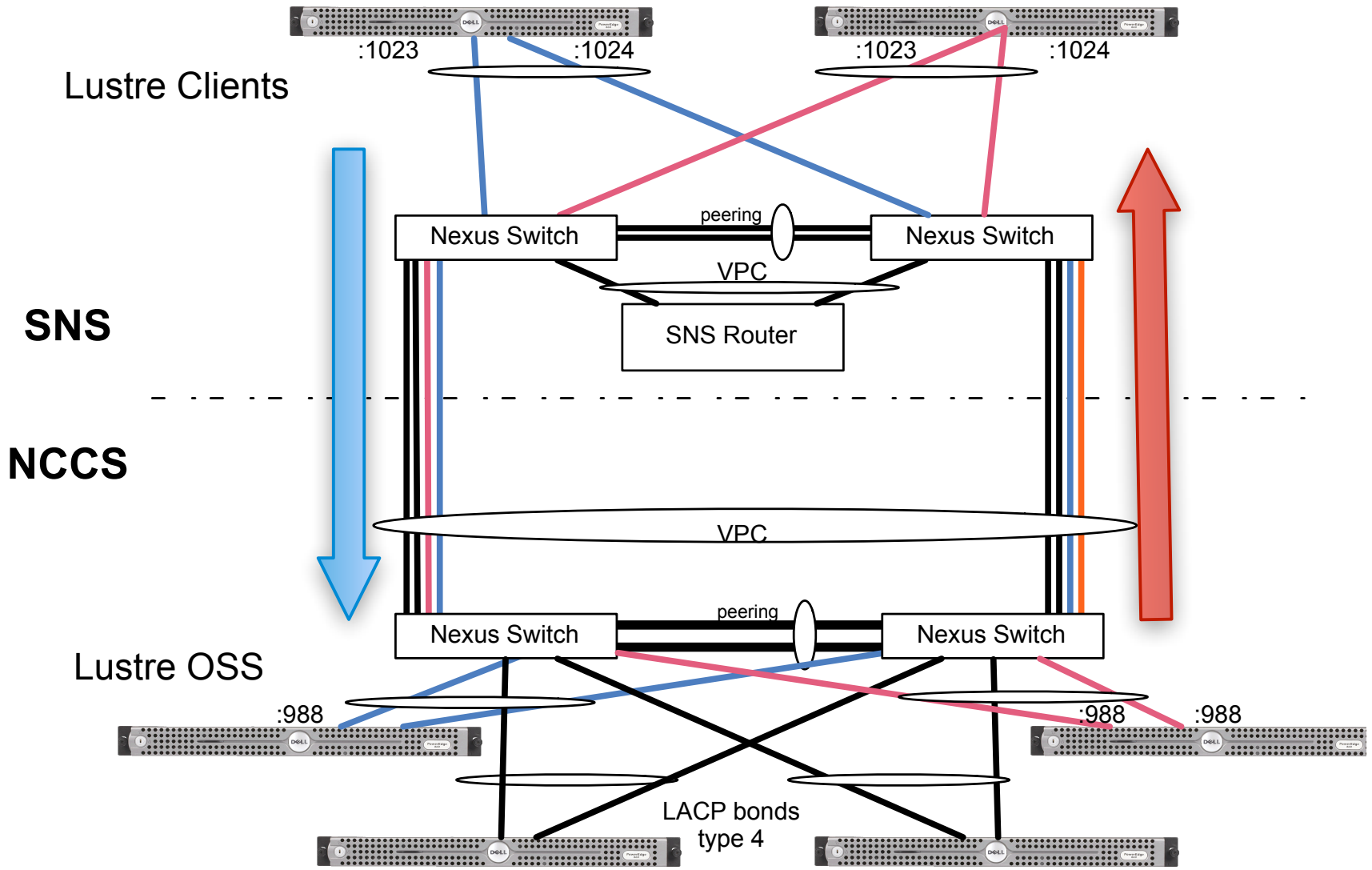
SOCKLND for HPC?

- Compatible with existing infrastructures (LAN/WAN)
- Converged fabric (management Eth, IPMI, LNET)
- Access control through host and network L3 filtering
- Channel-bonding in Linux kernel
  - Remove single points of failure with LNET
  - Disadvantage: increased complexity

Case Study (SNS Lustre filesystem)

- A Lustre deployment for Spallation Neutron Source at Oak Ridge National Laboratory
- 448TB, 4OSS/1MDS, Lustre 1.8, 2x10GE (channel-bonded), DDN SFA10K.
  - Backend is capable of 12GB/s (verified with xdd)
  - LNET capable of 8GB/s
- 1-2miles of fiber between SNS and NCCS (ORNL)

LACP Hashing



Network Validation

- Basic L1/L2 functionality testing and then try throughput test
  - Iperf/netperf for basic validation (e.g. more than 10Gb/s)
  - Testing for packet loss at 9Gb/s with UDP
    - iperf -w8m -u -l 16384 –c 10.x.x.x -b9G -i 2
- Complicated by redundant links
  - Had to “break” the port-channel bonds one-by-one
- 9K MTU clean path?
  - ping -s 8972 –Mdo 10.x.x.x

Latency Measurement

- Tools (ping, netperf, NetPIPE)
- Consider application latency as well
  - Different than hardware vendor's latency spec.
  - Without caching effects: NPtcp -I
- NetPIPE measurements (8192 byte messages)
  - 105us for SNS
  - 75us between OSS (2 switches)
  - 40us host-to-host
  - 20us IPoIB host-to-host
- Consider effects of interrupt coalescing

NIC Tuning

- Myricom Performance Tuning Guide
  - Interrupt binding/coalescing
- net.ipv4.tcp_sack = 0 (!!!)
  - Symptom was conflicting iperf tests sometimes 9Gb/s, then 1Gb/s. Repeatable, but independent of direction.
  - /etc/infiniband/openib.conf: `RUN_SYSCTL=yes`
    - /sbin/sysctl_perf_tuning (OFED 1.5.x)
- PCIe MaxPayload

```
# lspci -vv
MaxPayload 128 bytes, MaxReadReq 4096 bytes
```

TCP Host Kernel Parameters

- Sysctl parameters

```
# receive window
net.ipv4.tcp_no_metrics_save = 0
net.ipv4.tcp_window_scaling = 1
# congestion control
net.ipv4.tcp_congestion_control = htcp [cubic]
net.ipv4.tcp_timestamps = 0
# for ethernet networks
net.ipv4.tcp_sack = 1
```

- Good recommendations at http://fasterdata.es.net

SOCKLND

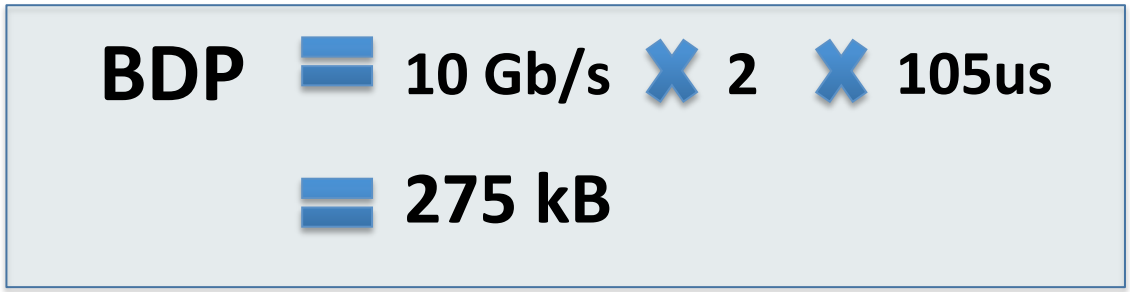
- Module parameters: credits, peer_credits, enable_irq_affinity
- Lctl conn_list
  - List active TCP connections, type (bulk/control), tx_buffer_size, rx_buffer_size

```
[root@sns-client ~]# lctl --net tcp conn_list
12345-128.219.249.38@tcp O[14]sns-client.ornl.gov->sns-oss4.ornl.gov:988 5863480/87380
nonagle
12345-128.219.249.38@tcp I[13]sns-client.ornl.gov->sns-oss4.ornl.gov:988 65536/87380
nonagle
12345-128.219.249.38@tcp C[9]sns-client.ornl.gov->sns-oss4.ornl.gov:988 65536/3350232
nonagle

[root@sns-oss4 ~]# lctl --net tcp conn_list|grep sns-client
12345-128.219.249.34@tcp I[2]sns-oss4.ornl.gov->sns-client.ornl.gov:1021 65536/16777216
nonagle
12345-128.219.249.34@tcp O[1]sns-oss4.ornl.gov->sns-client.ornl.gov:1022 65536/87380
nonagle
12345-128.219.249.34@tcp C[0]sns-oss4.ornl.gov->sns-client.ornl.gov:1023 65536/1492168
nonagle
[root@sns-oss4 ~]# netstat -tlpa|grep sns-mds2
tcp        0      0 sns-oss4.ornl.gov:988       sns-mds2.ornl.gov:1023      ESTABLISHED -
tcp        0      0 sns-oss4.ornl.gov:988       sns-mds2.ornl.gov:1022      ESTABLISHED -
tcp        0      0 sns-oss4.ornl.gov:988       sns-mds2.ornl.gov:1021      ESTABLISHED -
```

Lustre Parameters

- osc.*.checksums
  - Without checksums: single threaded writes up to 900MB/s
  - With checksums: 400-600MB/s
- osc.*.max_rpcs_in_flight
  - Increase for small IO or long fast network paths (high BDP)
  - May want to decrease to preempt TCP congestion



LNET Selftest

- lst add_test --concurrency [~max_rpcs_in_flight]
- lst add_test --distribute 1:1
  - expect 1150 MB/s out of each pair with concurrency
- lst add_test –distribute 1:4 --concurrency 8
  - Look for improvements from hashing across bonds
- lst add_test –distribute 4:1 --concurrency 8
  - Evaluate congestion control settings
- Use as a workload for packet header capture (tcpdump)
  - Congestion window sizing
  - Bandwidth efficiency - % of theoretical bw lost to TCP congestion avoidance

Observing Effects Tuning

- lst add_test --batch bw_test --loop 8192 --concurrency 1 --distribute 1:1 --from c --to s brw read size=1M

```
/proc/sys/lnet/peers:
nid                      refs state   max   rtr   min    tx   min queue
128.219.249.45@tcp          2    up     8     8     8     7     6 1048648

[LNet Rates of s]
[W] Avg: 1397     RPC/s Min: 1397     RPC/s Max: 1397     RPC/s
[LNet Bandwidth of s]
[W] Avg: 698.37   MB/s  Min: 698.37   MB/s  Max: 698.37   MB/s
```

- --concurrency 16

```
/proc/sys/lnet/peers:
128.219.249.45@tcp         15    up     8     8     8    -6    -9 11535824

LNet Rates of s]
[W] Avg: 2363     RPC/s Min: 2363     RPC/s Max: 2363     RPC/s
[LNet Bandwidth of s]
[W] Avg: 1181.56  MB/s  Min: 1181.56  MB/s  Max: 1181.56  MB/s
```

- options ksocklnd credits=4 peer_credits=2 (with --concurrency 3)

```
/proc/sys/lnet/nis:
nid                      status alive refs peer  rtr   max    tx   min
128.219.249.34@tcp           up    -1    1    2    0     4     4     4

/proc/sys/lnet/peers:
nid                      refs state   max   rtr   min    tx   min queue
128.219.249.45@tcp          4    up     2     2     2    -1    -2 3145944
```

Results with 2x10GE

- 2.1 GB/s sequential writes with fio (6 threads, file per thread)
- 1.58 GB/s cache to disk file copy (using NASA's mcp)
  - Options: --direct-read --direct-write --double-buffer --threads=4 –buffer-size=128
- 900 MB/s with dd
  - Lustre checksums off, MaxPayload=256

Questions?