CERN openlab for DataGrid Applications

A. Hirstius, S. Jarp

April 2003

Preliminary test results of the 10 GB NICs in back-to-back connections.

The 2.5.67 developer kernel with IA64 patches and the ixgb-driver version 1.0.47 are used.

Troughput test with different settings of internal kernel parameters and parameters of the driver.
The program "gensink" was used for testing the transfer rate. A very simple program, opening a TCP connection and writing data. In an earlier test iperf showed similar transfer rates, but is much"slower"in delivering the numbers. The final numbers will be, of course, measured with either iperf or netperf.
The measurements show fluctuations of of a few %.

Set 1) Transfer rate with the default settings (see Appendix).

MTU	***1 Stream***	***4 Streams***	***8 Streams***	***12 Streams***
1500	127 MB/s	4x94MB/s(tot:375)	8x65MB/s (tot:523)	12x44MB/s (tot:523)
9000	173 MB/s	4x91MB/s(tot:364)	8x77MB/s (tot:615)	12x58MB/s (tot:698)
16114	197 MB/s	4x100MB/s(tot:401)	8x71MB/s (tot:567)	12x53MB/s (tot:636)

Set 2) Transfer rate with improved kernel parameters:

MTU	***1 Stream***	***4 Streams***	***8 Streams***	***12 Streams***
1500	203 MB/s	4x103MB/s(tot:415)	8x57MB/s (tot:457)	12x41MB/s (tot:497)
9000	329 MB/s	4x151MB/s(tot:604)	8x79MB/s (tot:632)	12x55MB/s (tot:662)
16114	399 MB/s	4x179MB/s(tot:614)	8x86MB/s (tot:688)	12x59MB/s (tot:**712**)

Set 3) Transfer rate with improved kernel and driver parameters:

MTU	***1 Stream***	***4 Streams***	***8 Streams***	***12 Streams***
1500	275 MB/s	4x83MB/s (tot:331)	8x41MB/s (tot:329)	12x25MB/s (tot:295)
9000	693 MB/s	4x171MB/s(tot:685)	8x80MB/s (tot:642)	12x54MB/s (tot:643)
16114	**755 MB/s**	4x187MB/s(tot:**749**)	8x88MB/s (tot:**705**)	12x58MB/s (tot:698)

The CPU usage during all tests in set 2 and is on the sink side always about 100%!

The NIC was placed in PCI-X slot 4 and it turned out to run only at half speed.
The transfer rate was about 470 MB/s for single stream with optimal parameters, while the CPU usage was around 60% on the sink side. After placing the NIC in

PCI-X slot 1 the transfer rate went up to 750 MB/s with 100% CPU usage. With a faster CPU it should be, in principle, possible to reach twice the transfer rate of the half speed slot 470 MB/s => 940 MB/s.

For the current setup an aggregate transfer rate around 700 MB/s for multiple streams seams to be the limit, which is already reached with "sub-optimal" parameter settings.

For the tests with 12 streams the driver settings, that improve the rate for a small number of streams, reduce the aggregate rate.

Preliminary conclusions:

Important changes:

- Larger MTU settings especially in Set 3.
- Driver parameters, especially lowering RxIntDelay (for small number of connections)
- Increasing the settings for memory sizes in the kernel (mem, rmem, wmem)
- Use of PCI-X slot 1 (full speed) instead of slot 4 (half speed) vital

Changes with less impact (only few %):

- the remaining kernel parameters
- Driver parameters RxDescriptors and TxDescriptors (if not smaller than default)
- The MMRBC PCI-X parameter

Multiple streams are very important and have a relatively stable and high transfer rate with default settings.
With default MTU (1500) and "improved" parameters the transfer rate degrades!

Future work:

- Repeat the tests with tuned Enterasys switch
- Test with a fast IA-32 system (Xeon) with full speed (133 Mhz/64bit) PCI-X to be able to run without CPU limitations
- Test cross-combinations IA-32 and IA-64
- Test with improved IA-64 processors (Madison upgrade) to escape CPU limitations
- Repeat test with iperf or netperf
- more playing around with parameters

The default settings:

driver parameters:

RxIntDelay = 64 (interrupt delay in 0.8192 microsec units)
RxDescriptors=1024
TxDescriptors=256
TSO is by default ON if available(!!)

Kernel parameters:

net.ipv4.tcp_rmem="4096 87380 174760"
net.ipv4.tcp_wmem="4096 16384 131072"
net.ipv4.tcp_mem="97280 97792 98304"
net.core.netdev_max_backlog=300
net.core.rmem_default=65535
net.core.wmem_default=65535
net.core.rmem_max=65535
net.core.wmem_max=65535
net.core.optmem_max=20480
net.ipv4.tcp_sack=1
net.ipv4.tcp_timestamps=1
net.ipv4.tcp_tw_recycle=0
net.ipv4.tcp_tw_reuse=0

Improved kernel parameters:

net.ipv4.tcp_sack=0
net.ipv4.tcp_timestamps=0
net.core.rmem_default=524287
net.core.wmem_default=524287
net.core.rmem_max=524287
net.core.wmem_max=524287
net.core.optmem_max=524287
net.core.netdev_max_backlog=300000
net.ipv4.tcp_rmem="10000000 10000000 10000000"
net.ipv4.tcp_wmem="10000000 10000000 10000000"
net.ipv4.tcp_mem="10000000 10000000 10000000"
net.ipv4.tcp_tw_recycle=1
net.ipv4.tcp_tw_reuse=1

Improved driver parameters:

RxIntDelay = 0 (interrupt delay in 0.8192 microsec units)
RxDescriptors=2048
TxDescriptors=2048