Dual Processor Nodes for HPC Clusters: Network Connections

Douglas Eadline, Ph.D.

16th April 2003

deadline@basement-supercomputing.com

Abstract

Tests were run to determine the effect of sharing an Ethernet interface on a dual SMP node. The tests, as expected, indicate interface contention on the shared interface. Both Netperf and Netpipe tests were run on three different motherboard/chipset systems. The use of two Ethernet interfaces with the NAS test suite was explored. The results indicate dual SMP systems can benefit from multiple interfaces.

Background

Previous articles [1],[2] presented memory contention data for SMP motherboards. In this article we present some tests using Ethernet interfaces. The main question to be investigated is the effect of sharing the interface between two processors. The obvious answer is that the Ethernet channel capacity is reduced, but some quantitative measure should give some insight into the dynamics of these systems. Indeed, a follow-on question to consider is the effect of using two interfaces on a dual system.

Test Method and Hardware

In order to test the efficiency of dual nodes we can employ the Netperf and Netpipe benchmarks. These two benchmarks are in the Beowulf Performance Suite (BPS) `http://www.plogic.com/bps`. We used three systems:

- Tyan 2762 with dual Athlon MP 1600+ (1.4Ghz) processors, 2 GByte of RAM, dual 3Com 3c59x interfaces, running kernel 2.4.20
- Supermicro P3TDL3 with dual Tualatin 1.26 Ghz processors, 2 GByte of RAM, dual Intel EEPro100 interfaces, running kernel 2.4.18
- Intel SE7500WV2 with dual Xeon 2.2Ghz processors, 1 GByte of RAM, dual Intel EEPro100 interfaces, running kernel 2.4.20

The systems were arranged so that each node under test connected was two receiver nodes. The receiver nodes were dual PIII-800 systems with Intel EEpro100 interfaces.

A simple script has been written to run the tests as follow.

1. Run Netperf and Netpipe on one interface. This configuration is intended to simulate one processes on the same node using a single interface.

2. Run two copies of Netperf and Netpipe using the same interface sending to two different receiver nodes. This configuration is intended to simulate two processes on the same node using the same interface.

3. Run two copies of Netperf and Netpipe using two interfaces sending to two different receiver nodes.

Finally, the effect of using two interfaces for the NAS test suite was studied. A small 4 node (dual PIII-800) cluster was used for these tests.

Results

Supermicro P3TDL3

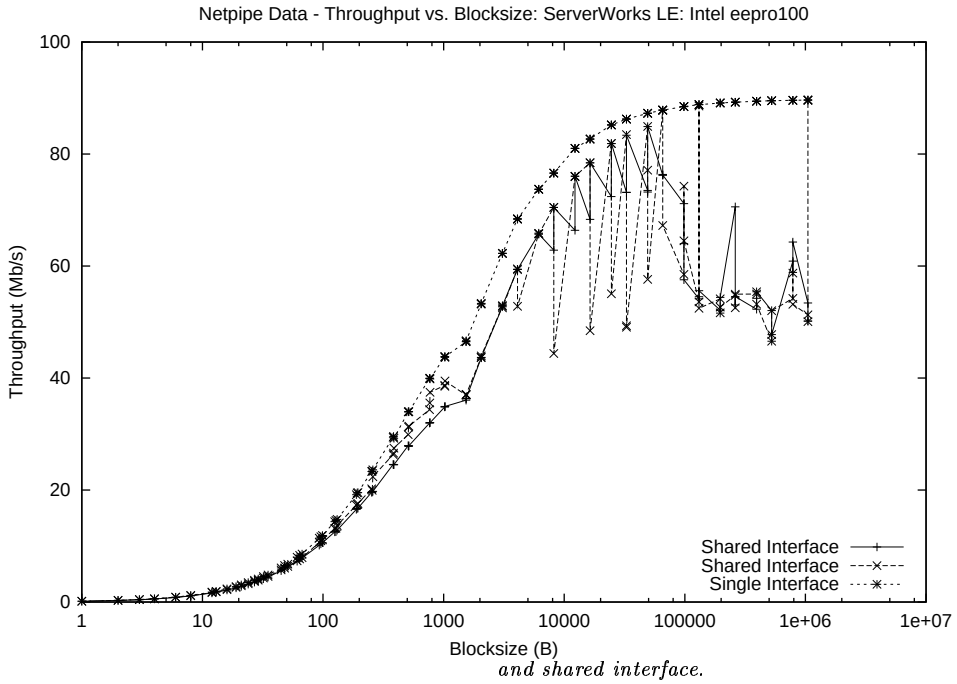
Netperf Test

The following table indicates the Netperf results:

test	UDP Mbits/sec	TCP Mbits/sec
Single test on single interface	95.70	94.13
Two tests on single Interface	47.86/47.86	46.99/47.15
Two tests on two interfaces	95.70/95.70	94.13/94.14

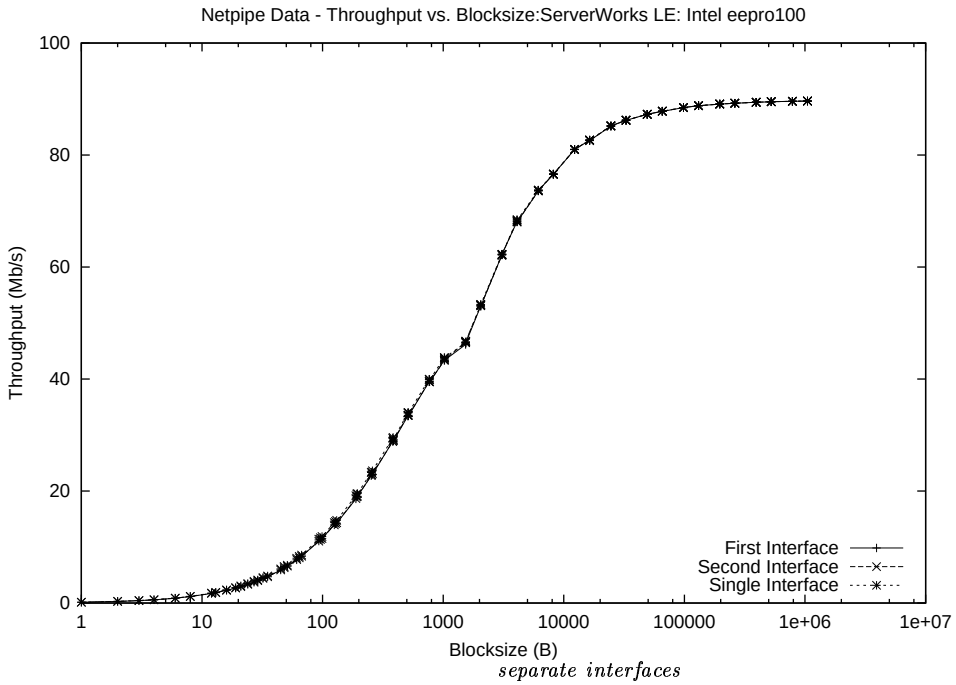
Netpipe Test

The following graph shows three sets of results. A single interface running the Netpipe benchmark (i.e. test 1 above). The same interface running two Netpipe benchmarks to two different nodes. (i.e. test 2 above). The vertical line at the end of the graph is the jump back to full bandwidth as one of the shared tests finished first.



Netpipe results for single and shared interface.

The following graph is Netpipe run on two separate interfaces connected with two separate receivers (e.g. test 3 above).



Netpipe results for two separate interfaces

Tyan 2762

Netperf Test

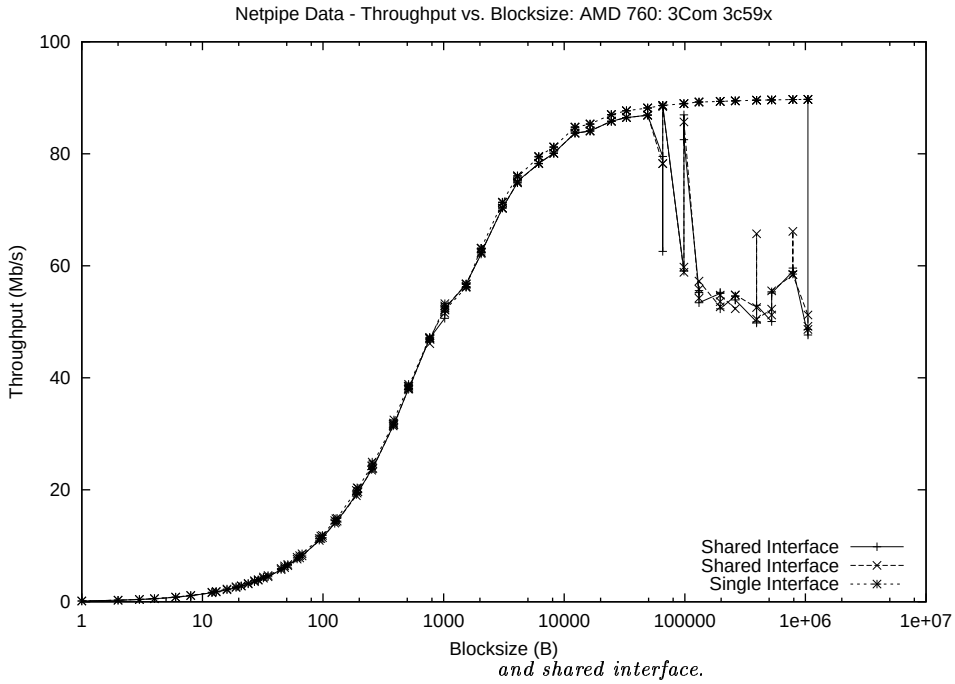
The following table indicates the Netperf results (* = invalid result, program seemed to have some math error when computing rate):

test	UDP Mbits/sec	TCP Mbits/sec
Single test on single interface	95.72	94.15
Two tests on single Interface	*	47.26/46.89
Two tests on two interfaces	95.73/95.73	94.16/94.16

Netpipe Test

The following graph shows three sets of results. A single interface running the Netpipe benchmark (i.e. test 1 above). The same interface running two Netpipe benchmarks to two different nodes. (i.e. test 2 above). The vertical line at the end of the graph is the jump back to full bandwidth as one of the shared tests finished first.

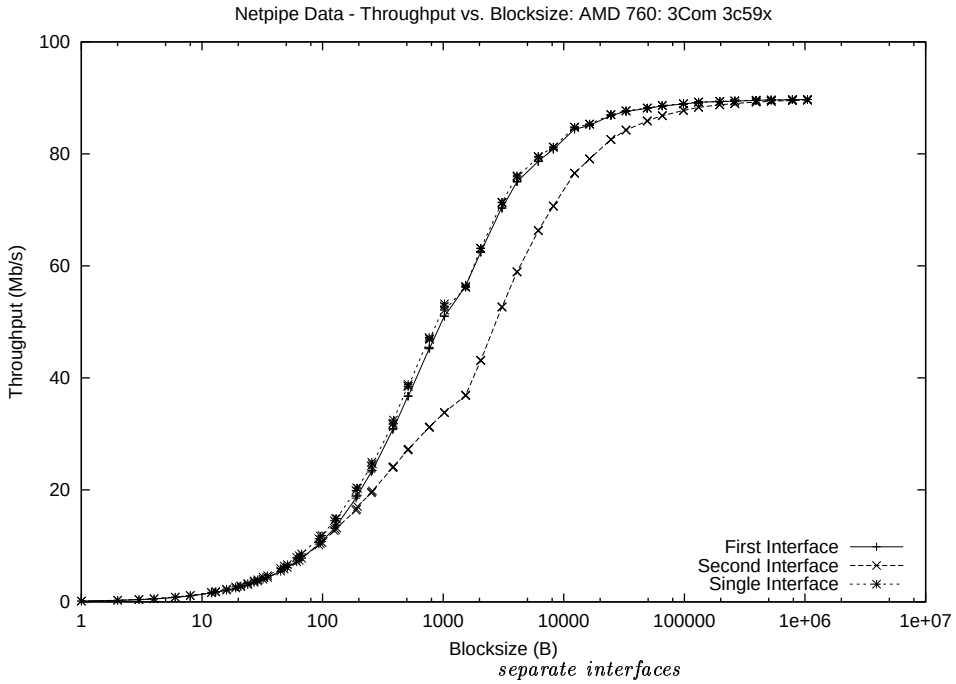
Netpipe results for single



and shared interface.

The following graph is Netpipe run on two separate interfaces connected with two separate receivers (e.g. test 3 above).

Netpipe results for two



separate interfaces

Intel SE7500WV2

Netperf Test

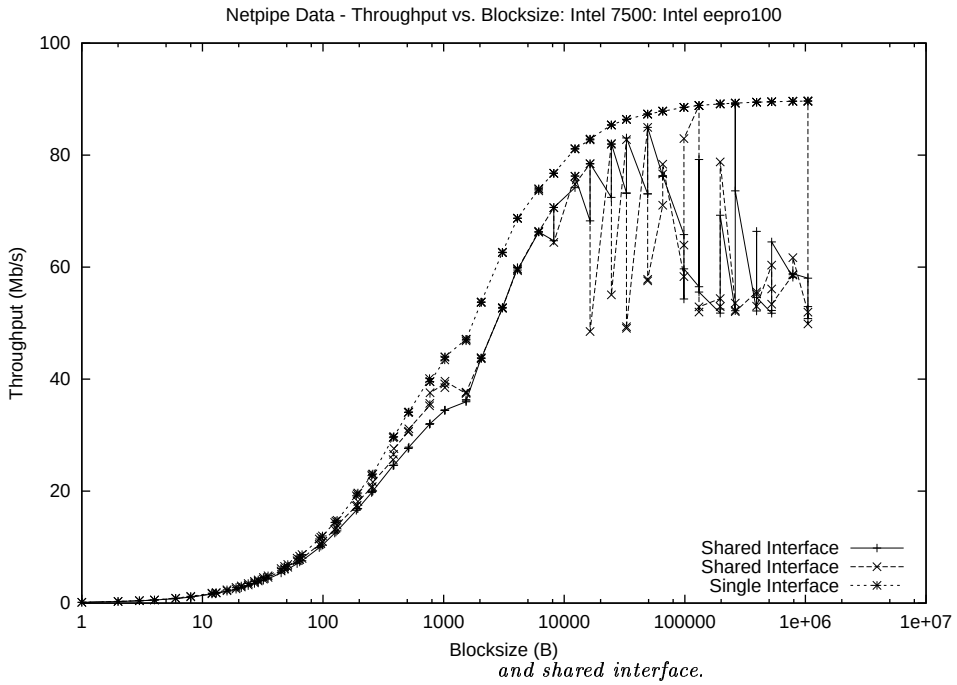
The following table indicates the Netperf results:

test	UDP Mbits/sec	TCP Mbits/sec
Single test on single interface	95.72	94.15
Two tests on single Interface	47.87/47.86	47.07/47.07
Two tests on two interfaces	95.70/95.71	94.15/94.12

Netpipe Test

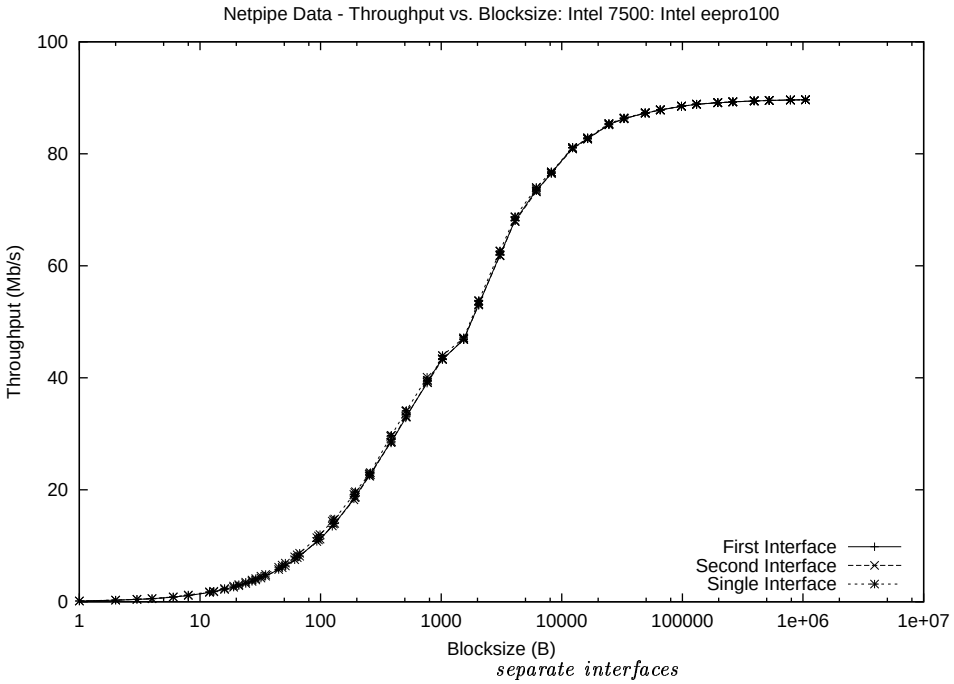
The following graph shows three sets of results. A single interface running the Netpipe benchmark (single Interface). The same interface running two Netpipe benchmarks to two different nodes. (i.e. there were three nodes in the test.) The vertical line at the end of the graph is the jump back to full bandwidth as one of the shared tests finished first.

Netpipe results for single and shared interface.



The following graph is Netpipe run on two separate interfaces connected with two separate receivers (e.g. test 3 above).

Netpipe results for two separate interfaces



The Effect of Two Interfaces on MPI jobs

As many motherboards have at lest two 100/100 Ethernet interfaces, it is interesting to see the effect of using both of these interfaces as separate networks for an MPI job. This test is not channel bonding were two interfaces are combined to form a single interface. To tun these tests, a small cluster with four dual processor nodes (PIII-800) and dual Ethernet (EEPro100) was used. The systems was configured to have two separate networks (192.168.0.0, 192.168.1.0). The /etc/hosts file was altered to show the same systems but though a different net. For instance:

```
norfolk1 192.168.0.1
norfolk2 192.168.0.2
norfolk3 192.168.0.3
norfolk4 192.168.0.4
norfolk1a 192.168.1.1
norfolk2a 192.168.1.2
norfolk3a 192.168.1.3
norfolk4a 192.168.1.4
```

The machines file uses for MPICH where as follows. For the a four CPU test (single interface):

```
norfolk1
norfolk2
norfolk3
norfolk4
```

For an eight CPU test (single interface):

```
norfolk1
norfolk2
norfolk3
norfolk4
```

For an eight CPU test (dual interfaces)

```
norfolk1
norfolk2
norfolk3
norfolk4
norfolk1a
norfolk2a
norfolk3a
norfolk4a
```

Running this test using LAM is a bit more difficult as multiple LAM daemons must be run on the same node. The total Mop/s for each are given below. (* = test did not run because it require square number of CPUs)

test	1 CPU	4 CPUs	8 CPUs Shared Interface	8 CPUs Dual Interfaces
BT	77.66	270.55	*	*
CG	54.80	116.62	80.91	132.94
EP	2.93	11.52	21.88	21.95
FT	124.49	111.53	136.37	152.39
IS	7.77	4.01	4.85	5.27
LU	100.20	358.64	399.63	422.81
SP	71.22	210.49	287.07	306.74
MG	51.96	158.93	*	*

The speed-up are given below (speedup=parallel Mops/sequential Mops) is provide below.

test	1 CPU	4 CPUs	8 CPUs Shared Interface	8 CPUs Dual Interfaces
BT	1	3.48	*	*
CG	1	2.13	1.48	2.43
EP	1	3.93	7.47	7.49
FT	1	.9	1.10	1.22
IS	1	.52	.62	.68
LU	1	3.58	3.99	4.22
SP	1	2.96	4.03	4.31
MG	1	3.06	*	*

Discussion

Before discussing the results, it should be mentioned that the excellent performance of Linux Ethernet is due to the drivers written by Don Becker of Scyld Computing. The Netperf and Netpipe results are what one would expect. Sharing the interface reduces the bandwidth available to an application. Perhaps the most interesting result is the Tyan 2762 does seem to keep pace with single interface performance until it

reaches larger bloc sizes. More interestingly, the Tyan 2762 shows curious behavior when the two interfaces are communicating separately. In the two other cases, the two Netpipe jobs showed virtually identical performance to the single job/single interface test.

The MPICH test is a bit more telling. The tests were run for 1, 4 and 8 CPU cases. In the the 8 CPU case, an extra interface was employed to see if contention on one interface had any effect on the tests. In general the answer seem to be that two interfaces on a dual node are better than one. With exception of the EP (embarrassing parallel) test, all the other tests, have a high rate of communication and seem to benefit from the extra interface. Of note is the poor performance of IS (integer sort) in all cases. This test is a highly latency sensitive test and even on 8 CPUs with two interfaces can not reach the single CPU performance.

The tests indicate the dual SMP nodes with a single network connection generate contention on the interface and therefore lower performance. Using two separate interfaces seems to help the performance for some of the NAS suite tests reinforcing the claim that sharing a single interface can reduce performance for certain applications. There does not appear to be many clusters that have utilized two interfaces as separate communication channels as a mean to increase performance. Of course, the addition of an extra interface is also highly dependent on the applications communication behavior.

Test Script

```
#!/bin/bash
# Script to test communication
# Usage: "run host1 host2 host3"
# host1 and host2 are on same net (interface)
# host 3 is on different net
#Run netperf between nodes
#make sure netserver is running on host2 and host3
date |tee net-test.out
sleep 3
echo "SENDING NETPERF TO ONE HOST" |tee -a net-test.out
./netperf -t UDP_STREAM -n 2 -l 180 -H $1 -- -s 65535 -m 1472 |tee -a net-test.out
./netperf -t TCP_STREAM -n 2 -l 180 -H $1 |tee -a net-test.out
echo "SENDING NETPERF TO TWO HOSTS SAME NET" |tee -a net-test.out
./netperf -t UDP_STREAM -n 2 -l 180 -H $1 -- -s 65535 -m 1472 |tee -a net-test.out &
./netperf -t UDP_STREAM -n 2 -l 180 -H $2 -- -s 65535 -m 1472 |tee -a net-test.out
./netperf -t TCP_STREAM -n 2 -l 180 -H $2 |tee -a net-test.out &
./netperf -t TCP_STREAM -n 2 -l 180 -H $1 |tee -a net-test.out
echo "SENDING NETPERF TO TWO HOSTS DIFFERENT NET" |tee -a net-test.out
./netperf -t UDP_STREAM -n 2 -l 180 -H $1 -- -s 65535 -m 1472 |tee -a net-test.out &
./netperf -t UDP_STREAM -n 2 -l 180 -H $3 -- -s 65535 -m 1472 |tee -a net-test.out
./netperf -t TCP_STREAM -n 2 -l 180 -H $3 |tee -a net-test.out &
./netperf -t TCP_STREAM -n 2 -l 180 -H $1 |tee -a net-test.out
#Now run Netpipe
rsh $1 net/NPtcp -r &
sleep 30
echo "SENDING NPtcp TO ONE HOST" |tee -a net-test.out
./NPtcp -t -h $1 -o netpipe.out.single -u 1048576
sleep 3
rsh $1 net/NPtcp -r &
rsh $2 net/NPtcp -r &
sleep 30
echo "SENDING NPtcp TO TWO HOSTS SAME NET" |tee -a net-test.out
./NPtcp -t -h $1 -o netpipe.out.dual1-same -u 1048576 &
./NPtcp -t -h $2 -o netpipe.out.dual2-same -u 1048576
sleep 3
echo "SENDING NPtcp TO TWO HOSTS DIFF NET" |tee -a net-test.out
sleep 3
rsh $1 net/NPtcp -r &
```

```
rsh $3 net/NPtcp -r &
sleep 30
./NPtcp -t -h $1 -o netpipe.out.dual1-diff -u 1048576 &
./NPtcp -t -h $3 -o netpipe.out.dual2-diff -u 1048576
```

References

[1] Dual Processor Nodes for HPC Clusters: Memory Contention Issues, Douglas Eadline, http://www.hpc-design.com/reports/smp-mem1/index.html

[2] More Dual Processor Nodes for HPC Clusters: Memory Contention Issues, Douglas Eadline, http://www.hpc-design.com/reports/smp-mem2/index.html

Acknowledgments

I wish to acknowledge all the authors of the tests suites used in this package.

Copyright